GiNaC

1.8.10

Generated by Doxygen 1.18.0

1 Directory Hierarchy 1
1.1 Directories . . . 1

2 Namespace Index 5
2.1 Namespace List . . . 5

3 Hierarchical Index 7
3.1 Class Hierarchy . . . 7

4 Class Index 11
4.1 Class List . . . 11

5 File Index 19
5.1 File List . . . 19

6 Directory Documentation 23
6.1 ginac Directory Reference . . . 23

7 Namespace Documentation 29
7.1 GiNaC Namespace Reference . . . 29
7.1.1 Typedef Documentation . . . 68
7.1.1.1 archive_node_id . . . 68
7.1.1.2 archive_atom . . . 68
7.1.1.3 synthesize_func . . . 68
7.1.1.4 unarchive_map_t . . . 69
7.1.1.5 exvector . . . 69
7.1.1.6 exset . . . 69
7.1.1.7 exmap . . . 69
7.1.1.8 evalffunctype . . . 69
7.1.1.9 FUNCP_1P . . . 69
7.1.1.10 FUNCP_2P . . . 69
7.1.1.11 FUNCP_CUBA . . . 69
7.1.1.12 epvector . . . 69
7.1.1.13 epp . . . 70
7.1.1.14 exprseq . . . 70
7.1.1.15 paramset . . . 70
7.1.1.16 eval_funcp . . . 70
7.1.1.17 evalf_funcp . . . 70
7.1.1.18 conjugate_funcp . . . 70
7.1.1.19 real_part_funcp . . . 70
7.1.1.20 imag_part_funcp . . . 70
7.1.1.21 expand_funcp . . . 70
7.1.1.22 derivative_funcp . . . 70
7.1.1.23 expl_derivative_funcp . . . 71

7.1.1.24 power_funcp . . . 71
7.1.1.25 series_funcp . . . 71
7.1.1.26 print_funcp . . . 71
7.1.1.27 info_funcp . . . 71
7.1.1.28 eval_funcp_1 . . . 71
7.1.1.29 evalf_funcp_1 . . . 71
7.1.1.30 conjugate_funcp_1 . . . 71
7.1.1.31 real_part_funcp_1 . . . 71
7.1.1.32 imag_part_funcp_1 . . . 71
7.1.1.33 expand_funcp_1 . . . 72
7.1.1.34 derivative_funcp_1 . . . 72
7.1.1.35 expl_derivative_funcp_1 . . . 72
7.1.1.36 power_funcp_1 . . . 72
7.1.1.37 series_funcp_1 . . . 72
7.1.1.38 print_funcp_1 . . . 72
7.1.1.39 info_funcp_1 . . . 72
7.1.1.40 eval_funcp_2 . . . 72
7.1.1.41 evalf_funcp_2 . . . 72
7.1.1.42 conjugate_funcp_2 . . . 72
7.1.1.43 real_part_funcp_2 . . . 73
7.1.1.44 imag_part_funcp_2 . . . 73
7.1.1.45 expand_funcp_2 . . . 73
7.1.1.46 derivative_funcp_2 . . . 73
7.1.1.47 expl_derivative_funcp_2 . . . 73
7.1.1.48 power_funcp_2 . . . 73
7.1.1.49 series_funcp_2 . . . 73
7.1.1.50 print_funcp_2 . . . 73
7.1.1.51 info_funcp_2 . . . 73
7.1.1.52 eval_funcp_3 . . . 73
7.1.1.53 evalf_funcp_3 . . . 74
7.1.1.54 conjugate_funcp_3 . . . 74
7.1.1.55 real_part_funcp_3 . . . 74
7.1.1.56 imag_part_funcp_3 . . . 74
7.1.1.57 expand_funcp_3 . . . 74
7.1.1.58 derivative_funcp_3 . . . 74
7.1.1.59 expl_derivative_funcp_3 . . . 74
7.1.1.60 power_funcp_3 . . . 74
7.1.1.61 series_funcp_3 . . . 74
7.1.1.62 print_funcp_3 . . . 74
7.1.1.63 info_funcp_3 . . . 75
7.1.1.64 eval_funcp_4 . . . 75
7.1.1.65 evalf_funcp_4 . . . 75

7.1.1.66 conjugate_funcp_4 . . . 75
7.1.1.67 real_part_funcp_4 . . . 75
7.1.1.68 imag_part_funcp_4 . . . 75
7.1.1.69 expand_funcp_4 . . . 75
7.1.1.70 derivative_funcp_4 . . . 75
7.1.1.71 expl_derivative_funcp_4 . . . 75
7.1.1.72 power_funcp_4 . . . 75
7.1.1.73 series_funcp_4 . . . 76
7.1.1.74 print_funcp_4 . . . 76
7.1.1.75 info_funcp_4 . . . 76
7.1.1.76 eval_funcp_5 . . . 76
7.1.1.77 evalf_funcp_5 . . . 76
7.1.1.78 conjugate_funcp_5 . . . 76
7.1.1.79 real_part_funcp_5 . . . 76
7.1.1.80 imag_part_funcp_5 . . . 76
7.1.1.81 expand_funcp_5 . . . 76
7.1.1.82 derivative_funcp_5 . . . 77
7.1.1.83 expl_derivative_funcp_5 . . . 77
7.1.1.84 power_funcp_5 . . . 77
7.1.1.85 series_funcp_5 . . . 77
7.1.1.86 print_funcp_5 . . . 77
7.1.1.87 info_funcp_5 . . . 77
7.1.1.88 eval_funcp_6 . . . 77
7.1.1.89 evalf_funcp_6 . . . 77
7.1.1.90 conjugate_funcp_6 . . . 77
7.1.1.91 real_part_funcp_6 . . . 78
7.1.1.92 imag_part_funcp_6 . . . 78
7.1.1.93 expand_funcp_6 . . . 78
7.1.1.94 derivative_funcp_6 . . . 78
7.1.1.95 expl_derivative_funcp_6 . . . 78
7.1.1.96 power_funcp_6 . . . 78
7.1.1.97 series_funcp_6 . . . 78
7.1.1.98 print_funcp_6 . . . 78
7.1.1.99 info_funcp_6 . . . 78
7.1.1.100 eval_funcp_7 . . . 79
7.1.1.101 evalf_funcp_7 . . . 79
7.1.1.102 conjugate_funcp_7 . . . 79
7.1.1.103 real_part_funcp_7 . . . 79
7.1.1.104 imag_part_funcp_7 . . . 79
7.1.1.105 expand_funcp_7 . . . 79
7.1.1.106 derivative_funcp_7 . . . 79
7.1.1.107 expl_derivative_funcp_7 . . . 79

7.1.1.108 power_funcp_7 . . . 79
7.1.1.109 series_funcp_7 . . . 80
7.1.1.110 print_funcp_7 . . . 80
7.1.1.111 info_funcp_7 . . . 80
7.1.1.112 eval_funcp_8 . . . 80
7.1.1.113 evalf_funcp_8 . . . 80
7.1.1.114 conjugate_funcp_8 . . . 80
7.1.1.115 real_part_funcp_8 . . . 80
7.1.1.116 imag_part_funcp_8 . . . 80
7.1.1.117 expand_funcp_8 . . . 80
7.1.1.118 derivative_funcp_8 . . . 81
7.1.1.119 expl_derivative_funcp_8 . . . 81
7.1.1.120 power_funcp_8 . . . 81
7.1.1.121 series_funcp_8 . . . 81
7.1.1.122 print_funcp_8 . . . 81
7.1.1.123 info_funcp_8 . . . 81
7.1.1.124 eval_funcp_9 . . . 81
7.1.1.125 evalf_funcp_9 . . . 81
7.1.1.126 conjugate_funcp_9 . . . 81
7.1.1.127 real_part_funcp_9 . . . 82
7.1.1.128 imag_part_funcp_9 . . . 82
7.1.1.129 expand_funcp_9 . . . 82
7.1.1.130 derivative_funcp_9 . . . 82
7.1.1.131 expl_derivative_funcp_9 . . . 82
7.1.1.132 power_funcp_9 . . . 82
7.1.1.133 series_funcp_9 . . . 82
7.1.1.134 print_funcp_9 . . . 82
7.1.1.135 info_funcp_9 . . . 82
7.1.1.136 eval_funcp_10 . . . 83
7.1.1.137 evalf_funcp_10 . . . 83
7.1.1.138 conjugate_funcp_10 . . . 83
7.1.1.139 real_part_funcp_10 . . . 83
7.1.1.140 imag_part_funcp_10 . . . 83
7.1.1.141 expand_funcp_10 . . . 83
7.1.1.142 derivative_funcp_10 . . . 83
7.1.1.143 expl_derivative_funcp_10 . . . 83
7.1.1.144 power_funcp_10 . . . 83
7.1.1.145 series_funcp_10 . . . 84
7.1.1.146 print_funcp_10 . . . 84
7.1.1.147 info_funcp_10 . . . 84
7.1.1.148 eval_funcp_11 . . . 84
7.1.1.149 evalf_funcp_11 . . . 84

7.1.1.150 conjugate_funcp_11 . . . 84
7.1.1.151 real_part_funcp_11 . . . 84
7.1.1.152 imag_part_funcp_11 . . . 84
7.1.1.153 expand_funcp_11 . . . 84
7.1.1.154 derivative_funcp_11 . . . 85
7.1.1.155 expl_derivative_funcp_11 . . . 85
7.1.1.156 power_funcp_11 . . . 85
7.1.1.157 series_funcp_11 . . . 85
7.1.1.158 print_funcp_11 . . . 85
7.1.1.159 info_funcp_11 . . . 85
7.1.1.160 eval_funcp_12 . . . 85
7.1.1.161 evalf_funcp_12 . . . 85
7.1.1.162 conjugate_funcp_12 . . . 86
7.1.1.163 real_part_funcp_12 . . . 86
7.1.1.164 imag_part_funcp_12 . . . 86
7.1.1.165 expand_funcp_12 . . . 86
7.1.1.166 derivative_funcp_12 . . . 86
7.1.1.167 expl_derivative_funcp_12 . . . 86
7.1.1.168 power_funcp_12 . . . 86
7.1.1.169 series_funcp_12 . . . 86
7.1.1.170 print_funcp_12 . . . 87
7.1.1.171 info_funcp_12 . . . 87
7.1.1.172 eval_funcp_13 . . . 87
7.1.1.173 evalf_funcp_13 . . . 87
7.1.1.174 conjugate_funcp_13 . . . 87
7.1.1.175 real_part_funcp_13 . . . 87
7.1.1.176 imag_part_funcp_13 . . . 87
7.1.1.177 expand_funcp_13 . . . 87
7.1.1.178 derivative_funcp_13 . . . 88
7.1.1.179 expl_derivative_funcp_13 . . . 88
7.1.1.180 power_funcp_13 . . . 88
7.1.1.181 series_funcp_13 . . . 88
7.1.1.182 print_funcp_13 . . . 88
7.1.1.183 info_funcp_13 . . . 88
7.1.1.184 eval_funcp_14 . . . 88
7.1.1.185 evalf_funcp_14 . . . 88
7.1.1.186 conjugate_funcp_14 . . . 89
7.1.1.187 real_part_funcp_14 . . . 89
7.1.1.188 imag_part_funcp_14 . . . 89
7.1.1.189 expand_funcp_14 . . . 89
7.1.1.190 derivative_funcp_14 . . . 89
7.1.1.191 expl_derivative_funcp_14 . . . 89

7.1.1.192 power_funcp_14 . . . 89
7.1.1.193 series_funcp_14 . . . 89
7.1.1.194 print_funcp_14 . . . 90
7.1.1.195 info_funcp_14 . . . 90
7.1.1.196 eval_funcp_exvector . . . 90
7.1.1.197 evalf_funcp_exvector . . . 90
7.1.1.198 conjugate_funcp_exvector . . . 90
7.1.1.199 real_part_funcp_exvector . . . 90
7.1.1.200 imag_part_funcp_exvector . . . 90
7.1.1.201 expand_funcp_exvector . . . 90
7.1.1.202 derivative_funcp_exvector . . . 90
7.1.1.203 expl_derivative_funcp_exvector . . . 90
7.1.1.204 power_funcp_exvector . . . 91
7.1.1.205 series_funcp_exvector . . . 91
7.1.1.206 print_funcp_exvector . . . 91
7.1.1.207 info_funcp_exvector . . . 91
7.1.1.208 exhashmap . . . 91
7.1.1.209 spmap . . . 91
7.1.1.210 lookup_map . . . 91
7.1.1.211 lst . . . 91
7.1.1.212 uintvector . . . 91
7.1.1.213 unsignedvector . . . 91
7.1.1.214 exvectorvector . . . 92
7.1.1.215 sym_desc_vec . . . 92
7.1.1.216 digits_changed_callback . . . 92
7.1.1.217 print_context_class_info . . . 92
7.1.1.218 registered_class_info . . . 92
7.1.2 Enumeration Type Documentation . . . 92
7.1.2.1 anonymous enum . . . 92
7.1.3 Function Documentation . . . 93
7.1.3.1 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [1/34] . . . 93
7.1.3.2 GINAC_BIND_UNARCHIVER() [1/50] . . . 93
7.1.3.3 GINAC_DECLARE_UNARCHIVER() [1/52] . . . 93
7.1.3.4 write_unsigned() . . . 94
7.1.3.5 read_unsigned() . . . 94
7.1.3.6 operator<<() [1/16] . . . 94
7.1.3.7 operator<<() [2/16] . . . 94
7.1.3.8 operator>>() [1/3] . . . 95
7.1.3.9 operator>>() [2/3] . . . 95
7.1.3.10 find_factory_fcn() . . . 95
7.1.3.11 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [2/34] . . . 95
7.1.3.12 is_a() [1/3] . . . 96

7.1.3.13 is_exactly_a() [1/2] . . . 96
7.1.3.14 dynallocate() [1/2] . . . 97
7.1.3.15 dynallocate() [2/2] . . . 98
7.1.3.16 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [3/34] . . . 98
7.1.3.17 print_func< print_dflt >() [1/3] . . . 99
7.1.3.18 GINAC_BIND_UNARCHIVER() [2/50] . . . 99
7.1.3.19 GINAC_BIND_UNARCHIVER() [3/50] . . . 99
7.1.3.20 GINAC_BIND_UNARCHIVER() [4/50] . . . 99
7.1.3.21 GINAC_BIND_UNARCHIVER() [5/50] . . . 99
7.1.3.22 GINAC_BIND_UNARCHIVER() [6/50] . . . 99
7.1.3.23 GINAC_BIND_UNARCHIVER() [7/50] . . . 99
7.1.3.24 GINAC_BIND_UNARCHIVER() [8/50] . . . 100
7.1.3.25 is_dirac_slash() . . . 100
7.1.3.26 base_and_index() . . . 100
7.1.3.27 dirac_ONE() . . . 100
7.1.3.28 get_dim_uint() . . . 101
7.1.3.29 clifford_unit() . . . 101
7.1.3.30 dirac_gamma() . . . 101
7.1.3.31 dirac_gamma5() . . . 102
7.1.3.32 dirac_gammaL() . . . 102
7.1.3.33 dirac_gammaR() . . . 103
7.1.3.34 dirac_slash() . . . 103
7.1.3.35 get_representation_label() [1/2] . . . 103
7.1.3.36 trace_string() . . . 104
7.1.3.37 dirac_trace() [1/3] . . . 104
7.1.3.38 dirac_trace() [2/3] . . . 104
7.1.3.39 dirac_trace() [3/3] . . . 105
7.1.3.40 canonicalize_clifford() . . . 105
7.1.3.41 clifford_star_bar() . . . 106
7.1.3.42 clifford_prime() . . . 106
7.1.3.43 remove_dirac_ONE() . . . 106
7.1.3.44 clifford_max_label() . . . 107
7.1.3.45 clifford_norm() . . . 107
7.1.3.46 clifford_inverse() . . . 107
7.1.3.47 lst_to_clifford() [1/2] . . . 108
7.1.3.48 lst_to_clifford() [2/2] . . . 108
7.1.3.49 get_clifford_comp() . . . 109
7.1.3.50 clifford_to_lst() . . . 109
7.1.3.51 clifford_moebius_map() [1/2] . . . 110
7.1.3.52 clifford_moebius_map() [2/2] . . . 110
7.1.3.53 GINAC_DECLARE_UNARCHIVER() [2/52] . . . 111
7.1.3.54 GINAC_DECLARE_UNARCHIVER() [3/52] . . . 111

7.1.3.55 GINAC_DECLARE_UNARCHIVER() [4/52] . . . 111
7.1.3.56 GINAC_DECLARE_UNARCHIVER() [5/52] . . . 111
7.1.3.57 GINAC_DECLARE_UNARCHIVER() [6/52] . . . 111
7.1.3.58 GINAC_DECLARE_UNARCHIVER() [7/52] . . . 112
7.1.3.59 GINAC_DECLARE_UNARCHIVER() [8/52] . . . 112
7.1.3.60 is_clifford_tinfo() . . . 112
7.1.3.61 clifford_bar() . . . 112
7.1.3.62 clifford_star() . . . 112
7.1.3.63 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [4/34] . . . 113
7.1.3.64 print_func< print_dflt >() [2/3] . . . 113
7.1.3.65 GINAC_BIND_UNARCHIVER() [9/50] . . . 113
7.1.3.66 GINAC_BIND_UNARCHIVER() [10/50] . . . 113
7.1.3.67 GINAC_BIND_UNARCHIVER() [11/50] . . . 113
7.1.3.68 GINAC_BIND_UNARCHIVER() [12/50] . . . 113
7.1.3.69 GINAC_BIND_UNARCHIVER() [13/50] . . . 114
7.1.3.70 permute_free_index_to_front() . . . 114
7.1.3.71 color_ONE() . . . 114
7.1.3.72 color_T() . . . 115
7.1.3.73 color_f() . . . 115
7.1.3.74 color_d() . . . 116
7.1.3.75 color_h() . . . 116
7.1.3.76 is_color_tinfo() . . . 116
7.1.3.77 get_representation_label() [2/2] . . . 117
7.1.3.78 color_trace() [1/3] . . . 117
7.1.3.79 color_trace() [2/3] . . . 117
7.1.3.80 color_trace() [3/3] . . . 118
7.1.3.81 GINAC_DECLARE_UNARCHIVER() [9/52] . . . 118
7.1.3.82 GINAC_DECLARE_UNARCHIVER() [10/52] . . . 118
7.1.3.83 GINAC_DECLARE_UNARCHIVER() [11/52] . . . 118
7.1.3.84 GINAC_DECLARE_UNARCHIVER() [12/52] . . . 118
7.1.3.85 GINAC_DECLARE_UNARCHIVER() [13/52] . . . 118
7.1.3.86 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [5/34] . . . 119
7.1.3.87 GINAC_BIND_UNARCHIVER() [14/50] . . . 119
7.1.3.88 GINAC_DECLARE_UNARCHIVER() [14/52] . . . 119
7.1.3.89 crc32() . . . 119
7.1.3.90 are_ex_trivially_equal() . . . 119
7.1.3.91 operator<<() [3/16] . . . 120
7.1.3.92 operator<<() [4/16] . . . 120
7.1.3.93 operator<<() [5/16] . . . 120
7.1.3.94 nops() [1/2] . . . 120
7.1.3.95 expand() [1/2] . . . 120
7.1.3.96 conjugate() . . . 120

7.1.3.97 real_part() . . . 121
7.1.3.98 imag_part() . . . 121
7.1.3.99 has() . . . 121
7.1.3.100 find() . . . 121
7.1.3.101 is_polynomial() . . . 121
7.1.3.102 degree() . . . 122
7.1.3.103 ldegree() . . . 122
7.1.3.104 coeff() . . . 122
7.1.3.105 numer() [1/2] . . . 122
7.1.3.106 denom() [1/2] . . . 122
7.1.3.107 numer_denom() . . . 123
7.1.3.108 normal() . . . 123
7.1.3.109 to_rational() . . . 123
7.1.3.110 to_polynomial() . . . 123
7.1.3.111 collect() . . . 123
7.1.3.112 eval() . . . 124
7.1.3.113 evalf() [1/2] . . . 124
7.1.3.114 evalm() . . . 124
7.1.3.115 eval_integ() . . . 124
7.1.3.116 diff() . . . 124
7.1.3.117 series() . . . 125
7.1.3.118 match() . . . 125
7.1.3.119 simplify_indexed() [1/3] . . . 125
7.1.3.120 simplify_indexed() [2/3] . . . 125
7.1.3.121 symmetrize() [1/4] . . . 125
7.1.3.122 symmetrize() [2/4] . . . 126
7.1.3.123 antisymmetrize() [1/4] . . . 126
7.1.3.124 antisymmetrize() [2/4] . . . 126
7.1.3.125 symmetrize_cyclic() [1/4] . . . 126
7.1.3.126 symmetrize_cyclic() [2/4] . . . 126
7.1.3.127 op() . . . 126
7.1.3.128 lhs() . . . 127
7.1.3.129 rhs() . . . 127
7.1.3.130 is_zero() [1/2] . . . 127
7.1.3.131 swap() [1/2] . . . 127
7.1.3.132 subs() [1/3] . . . 127
7.1.3.133 subs() [2/3] . . . 128
7.1.3.134 subs() [3/3] . . . 128
7.1.3.135 is_a() [2/3] . . . 128
7.1.3.136 is_exactly_a() [2/2] . . . 128
7.1.3.137 ex_to() . . . 128
7.1.3.138 compile_ex() [1/3] . . . 130

7.1.3.139 compile_ex() [2/3] . . . 130
7.1.3.140 compile_ex() [3/3] . . . 131
7.1.3.141 link_ex() [1/2] . . . 131
7.1.3.142 link_ex() [2/2] . . . 132
7.1.3.143 unlink_ex() . . . 132
7.1.3.144 swap() [2/2] . . . 132
7.1.3.145 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [6/34] . . . 133
7.1.3.146 conjugateepvector() . . . 133
7.1.3.147 factor() . . . 133
7.1.3.148 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [7/34] . . . 134
7.1.3.149 GINAC_DECLARE_UNARCHIVER() [15/52] . . . 134
7.1.3.150 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [8/34] . . . 134
7.1.3.151 GINAC_BIND_UNARCHIVER() [15/50] . . . 134
7.1.3.152 GINAC_DECLARE_UNARCHIVER() [16/52] . . . 134
7.1.3.153 GINAC_BIND_UNARCHIVER() [16/50] . . . 134
7.1.3.154 GINAC_DECLARE_UNARCHIVER() [17/52] . . . 135
7.1.3.155 is_the_function() . . . 135
7.1.3.156 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [9/34] . . . 135
7.1.3.157 GINAC_BIND_UNARCHIVER() [17/50] . . . 135
7.1.3.158 GINAC_DECLARE_UNARCHIVER() [18/52] . . . 135
7.1.3.159 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [10/34] . . . 135
7.1.3.160 print_func< print_context >() . . . 136
7.1.3.161 GINAC_BIND_UNARCHIVER() [18/50] . . . 136
7.1.3.162 GINAC_BIND_UNARCHIVER() [19/50] . . . 136
7.1.3.163 GINAC_BIND_UNARCHIVER() [20/50] . . . 136
7.1.3.164 is_dummy_pair() [1/2] . . . 136
7.1.3.165 is_dummy_pair() [2/2] . . . 136
7.1.3.166 find_free_and_dummy() [1/2] . . . 137
7.1.3.167 minimal_dim() . . . 137
7.1.3.168 GINAC_DECLARE_UNARCHIVER() [19/52] . . . 137
7.1.3.169 GINAC_DECLARE_UNARCHIVER() [20/52] . . . 137
7.1.3.170 GINAC_DECLARE_UNARCHIVER() [21/52] . . . 138
7.1.3.171 find_free_and_dummy() [2/2] . . . 138
7.1.3.172 find_dummy_indices() . . . 138
7.1.3.173 count_dummy_indices() . . . 138
7.1.3.174 count_free_indices() . . . 139
7.1.3.175 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [11/34] . . . 139
7.1.3.176 GINAC_BIND_UNARCHIVER() [21/50] . . . 139
7.1.3.177 indices_consistent() . . . 139
7.1.3.178 number_of_type() . . . 139
7.1.3.179 rename_dummy_indices() . . . 140
7.1.3.180 find_variant_indices() . . . 140

7.1.3.181 reposition_dummy_indices() . . . 140
7.1.3.182 product_to_exvector() . . . 141
7.1.3.183 idx_symmetrization() . . . 141
7.1.3.184 simplify_indexed() [3/3] . . . 141
7.1.3.185 simplify_indexed_product() . . . 141
7.1.3.186 hasindex() . . . 141
7.1.3.187 get_all_dummy_indices_safely() . . . 142
7.1.3.188 get_all_dummy_indices() . . . 142
7.1.3.189 rename_dummy_indices_uniquely() [1/4] . . . 142
7.1.3.190 rename_dummy_indices_uniquely() [2/4] . . . 142
7.1.3.191 rename_dummy_indices_uniquely() [3/4] . . . 143
7.1.3.192 rename_dummy_indices_uniquely() [4/4] . . . 143
7.1.3.193 expand_dummy_sum() . . . 143
7.1.3.194 GINAC_DECLARE_UNARCHIVER() [22/52] . . . 143
7.1.3.195 conjugate_evalf() . . . 144
7.1.3.196 conjugate_eval() . . . 144
7.1.3.197 conjugate_print_latex() . . . 144
7.1.3.198 conjugate_conjugate() . . . 144
7.1.3.199 conjugate_expl_derivative() . . . 144
7.1.3.200 conjugate_real_part() . . . 145
7.1.3.201 conjugate_imag_part() . . . 145
7.1.3.202 func_arg_info() . . . 145
7.1.3.203 conjugate_info() . . . 145
7.1.3.204 REGISTER_FUNCTION() [1/36] . . . 146
7.1.3.205 real_part_evalf() . . . 146
7.1.3.206 real_part_eval() . . . 146
7.1.3.207 real_part_print_latex() . . . 146
7.1.3.208 real_part_conjugate() . . . 147
7.1.3.209 real_part_real_part() . . . 147
7.1.3.210 real_part_imag_part() . . . 147
7.1.3.211 real_part_expl_derivative() . . . 147
7.1.3.212 REGISTER_FUNCTION() [2/36] . . . 147
7.1.3.213 imag_part_evalf() . . . 148
7.1.3.214 imag_part_eval() . . . 148
7.1.3.215 imag_part_print_latex() . . . 148
7.1.3.216 imag_part_conjugate() . . . 148
7.1.3.217 imag_part_real_part() . . . 148
7.1.3.218 imag_part_imag_part() . . . 149
7.1.3.219 imag_part_expl_derivative() . . . 149
7.1.3.220 REGISTER_FUNCTION() [3/36] . . . 149
7.1.3.221 abs_evalf() . . . 149
7.1.3.222 abs_eval() . . . 149

7.1.3.223 abs_expand() . . . 150
7.1.3.224 abs_expl_derivative() . . . 150
7.1.3.225 abs_print_latex() . . . 150
7.1.3.226 abs_print_csrc_float() . . . 150
7.1.3.227 abs_conjugate() . . . 150
7.1.3.228 abs_real_part() . . . 151
7.1.3.229 abs_imag_part() . . . 151
7.1.3.230 abs_power() . . . 151
7.1.3.231 abs_info() . . . 151
7.1.3.232 REGISTER_FUNCTION() [4/36] . . . 151
7.1.3.233 step_evalf() . . . 152
7.1.3.234 step_eval() . . . 152
7.1.3.235 step_series() . . . 152
7.1.3.236 step_conjugate() . . . 152
7.1.3.237 step_real_part() . . . 152
7.1.3.238 step_imag_part() . . . 153
7.1.3.239 REGISTER_FUNCTION() [5/36] . . . 153
7.1.3.240 csgn_evalf() . . . 153
7.1.3.241 csgn_eval() . . . 153
7.1.3.242 csgn_series() . . . 153
7.1.3.243 csgn_conjugate() . . . 154
7.1.3.244 csgn_real_part() . . . 154
7.1.3.245 csgn_imag_part() . . . 154
7.1.3.246 csgn_power() . . . 154
7.1.3.247 REGISTER_FUNCTION() [6/36] . . . 154
7.1.3.248 eta_evalf() . . . 155
7.1.3.249 eta_eval() . . . 155
7.1.3.250 eta_series() . . . 155
7.1.3.251 eta_conjugate() . . . 155
7.1.3.252 eta_real_part() . . . 155
7.1.3.253 eta_imag_part() . . . 156
7.1.3.254 REGISTER_FUNCTION() [7/36] . . . 156
7.1.3.255 Li2_evalf() . . . 156
7.1.3.256 Li2_eval() . . . 156
7.1.3.257 Li2_deriv() . . . 156
7.1.3.258 Li2_series() [1/2] . . . 157
7.1.3.259 Li2_conjugate() . . . 157
7.1.3.260 REGISTER_FUNCTION() [8/36] . . . 157
7.1.3.261 Li3_eval() . . . 157
7.1.3.262 REGISTER_FUNCTION() [9/36] . . . 157
7.1.3.263 zetaderiv_eval() . . . 158
7.1.3.264 zetaderiv_deriv() . . . 158

7.1.3.265 REGISTER_FUNCTION() [10/36] . . . 158
7.1.3.266 factorial_evalf() . . . 158
7.1.3.267 factorial_eval() . . . 158
7.1.3.268 factorial_print_dflt_latex() . . . 159
7.1.3.269 factorial_conjugate() . . . 159
7.1.3.270 factorial_real_part() . . . 159
7.1.3.271 factorial_imag_part() . . . 159
7.1.3.272 REGISTER_FUNCTION() [11/36] . . . 159
7.1.3.273 binomial_evalf() . . . 160
7.1.3.274 binomial_sym() . . . 160
7.1.3.275 binomial_eval() . . . 160
7.1.3.276 binomial_conjugate() . . . 160
7.1.3.277 binomial_real_part() . . . 160
7.1.3.278 binomial_imag_part() . . . 161
7.1.3.279 REGISTER_FUNCTION() [12/36] . . . 161
7.1.3.280 Order_eval() . . . 161
7.1.3.281 Order_series() . . . 161
7.1.3.282 Order_conjugate() . . . 161
7.1.3.283 Order_real_part() . . . 162
7.1.3.284 Order_imag_part() . . . 162
7.1.3.285 Order_power() . . . 162
7.1.3.286 Order_expl_derivative() . . . 162
7.1.3.287 REGISTER_FUNCTION() [13/36] . . . 162
7.1.3.288 lsolve() . . . 163
7.1.3.289 fsolve() . . . 163
7.1.3.290 zeta() [1/3] . . . 164
7.1.3.291 zeta() [2/3] . . . 164
7.1.3.292 is_the_function< zeta_SERIAL >() . . . 164
7.1.3.293 G() [1/2] . . . 164
7.1.3.294 G() [2/2] . . . 164
7.1.3.295 is_the_function< G_SERIAL >() . . . 165
7.1.3.296 psi() [1/4] . . . 165
7.1.3.297 psi() [2/4] . . . 165
7.1.3.298 is_the_function< psi_SERIAL >() . . . 165
7.1.3.299 iterated_integral() [1/2] . . . 165
7.1.3.300 iterated_integral() [2/2] . . . 166
7.1.3.301 is_the_function< iterated_integral_SERIAL >() . . . 166
7.1.3.302 is_order_function() . . . 166
7.1.3.303 convert_H_to_Li() . . . 166
7.1.3.304 EllipticK_evalf() . . . 166
7.1.3.305 EllipticK_eval() . . . 167
7.1.3.306 EllipticK_deriv() . . . 167

7.1.3.307 EllipticK_series() . . . 167
7.1.3.308 EllipticK_print_latex() . . . 167
7.1.3.309 REGISTER_FUNCTION() [14/36] . . . 167
7.1.3.310 EllipticE_evalf() . . . 168
7.1.3.311 EllipticE_eval() . . . 168
7.1.3.312 EllipticE_deriv() . . . 168
7.1.3.313 EllipticE_series() . . . 168
7.1.3.314 EllipticE_print_latex() . . . 168
7.1.3.315 REGISTER_FUNCTION() [15/36] . . . 169
7.1.3.316 iterated_integral_evalf_impl() . . . 169
7.1.3.317 iterated_integral2_evalf() . . . 169
7.1.3.318 iterated_integral3_evalf() . . . 169
7.1.3.319 iterated_integral2_eval() . . . 169
7.1.3.320 iterated_integral3_eval() . . . 170
7.1.3.321 lgamma_evalf() . . . 170
7.1.3.322 lgamma_eval() . . . 170
7.1.3.323 lgamma_deriv() . . . 170
7.1.3.324 lgamma_series() . . . 171
7.1.3.325 lgamma_conjugate() . . . 171
7.1.3.326 REGISTER_FUNCTION() [16/36] . . . 171
7.1.3.327 tgamma_evalf() . . . 171
7.1.3.328 tgamma_eval() . . . 172
7.1.3.329 tgamma_deriv() . . . 172
7.1.3.330 tgamma_series() . . . 172
7.1.3.331 tgamma_conjugate() . . . 172
7.1.3.332 REGISTER_FUNCTION() [17/36] . . . 173
7.1.3.333 beta_evalf() . . . 173
7.1.3.334 beta_eval() . . . 173
7.1.3.335 beta_deriv() . . . 173
7.1.3.336 beta_series() . . . 173
7.1.3.337 REGISTER_FUNCTION() [18/36] . . . 174
7.1.3.338 psi1_evalf() . . . 174
7.1.3.339 psi1_eval() . . . 174
7.1.3.340 psi1_deriv() . . . 174
7.1.3.341 psi1_series() . . . 174
7.1.3.342 psi2_evalf() . . . 175
7.1.3.343 psi2_eval() . . . 175
7.1.3.344 psi2_deriv() . . . 175
7.1.3.345 psi2_series() . . . 175
7.1.3.346 G2_evalf() . . . 176
7.1.3.347 G2_eval() . . . 176
7.1.3.348 G3_evalf() . . . 176

7.1.3.349 G3_eval() . . . 176
7.1.3.350 Li_evalf() . . . 177
7.1.3.351 Li_eval() . . . 177
7.1.3.352 Li_series() . . . 177
7.1.3.353 Li_deriv() . . . 177
7.1.3.354 Li_print_latex() . . . 178
7.1.3.355 REGISTER_FUNCTION() [19/36] . . . 178
7.1.3.356 S_evalf() . . . 178
7.1.3.357 S_eval() . . . 178
7.1.3.358 S_series() . . . 179
7.1.3.359 S_deriv() . . . 179
7.1.3.360 S_print_latex() . . . 179
7.1.3.361 REGISTER_FUNCTION() [20/36] . . . 179
7.1.3.362 H_evalf() . . . 180
7.1.3.363 H_eval() . . . 180
7.1.3.364 H_series() . . . 180
7.1.3.365 H_deriv() . . . 180
7.1.3.366 H_print_latex() . . . 181
7.1.3.367 REGISTER_FUNCTION() [21/36] . . . 181
7.1.3.368 zeta1_evalf() . . . 181
7.1.3.369 zeta1_eval() . . . 181
7.1.3.370 zeta1_deriv() . . . 181
7.1.3.371 zeta1_print_latex() . . . 182
7.1.3.372 zeta2_evalf() . . . 182
7.1.3.373 zeta2_eval() . . . 182
7.1.3.374 zeta2_deriv() . . . 182
7.1.3.375 zeta2_print_latex() . . . 182
7.1.3.376 exp_evalf() . . . 183
7.1.3.377 exp_eval() . . . 183
7.1.3.378 exp_expand() . . . 183
7.1.3.379 exp_deriv() . . . 183
7.1.3.380 exp_real_part() . . . 183
7.1.3.381 exp_imag_part() . . . 184
7.1.3.382 exp_conjugate() . . . 184
7.1.3.383 exp_power() . . . 184
7.1.3.384 exp_info() . . . 184
7.1.3.385 REGISTER_FUNCTION() [22/36] . . . 184
7.1.3.386 log_evalf() . . . 185
7.1.3.387 log_eval() . . . 185
7.1.3.388 log_deriv() . . . 185
7.1.3.389 log_series() . . . 185
7.1.3.390 log_real_part() . . . 185

7.1.3.391 log_imag_part() . . . 186
7.1.3.392 log_expand() . . . 186
7.1.3.393 log_conjugate() . . . 186
7.1.3.394 log_info() . . . 186
7.1.3.395 REGISTER_FUNCTION() [23/36] . . . 186
7.1.3.396 sin_evalf() . . . 187
7.1.3.397 sin_eval() . . . 187
7.1.3.398 sin_deriv() . . . 187
7.1.3.399 sin_real_part() . . . 187
7.1.3.400 sin_imag_part() . . . 187
7.1.3.401 sin_conjugate() . . . 188
7.1.3.402 trig_info() . . . 188
7.1.3.403 REGISTER_FUNCTION() [24/36] . . . 188
7.1.3.404 cos_evalf() . . . 188
7.1.3.405 cos_eval() . . . 188
7.1.3.406 cos_deriv() . . . 189
7.1.3.407 cos_real_part() . . . 189
7.1.3.408 cos_imag_part() . . . 189
7.1.3.409 cos_conjugate() . . . 189
7.1.3.410 REGISTER_FUNCTION() [25/36] . . . 189
7.1.3.411 tan_evalf() . . . 190
7.1.3.412 tan_eval() . . . 190
7.1.3.413 tan_deriv() . . . 190
7.1.3.414 tan_real_part() . . . 190
7.1.3.415 tan_imag_part() . . . 190
7.1.3.416 tan_series() . . . 191
7.1.3.417 tan_conjugate() . . . 191
7.1.3.418 REGISTER_FUNCTION() [26/36] . . . 191
7.1.3.419 asin_evalf() . . . 191
7.1.3.420 asin_eval() . . . 191
7.1.3.421 asin_deriv() . . . 192
7.1.3.422 asin_conjugate() . . . 192
7.1.3.423 asin_info() . . . 192
7.1.3.424 REGISTER_FUNCTION() [27/36] . . . 192
7.1.3.425 acos_evalf() . . . 192
7.1.3.426 acos_eval() . . . 193
7.1.3.427 acos_deriv() . . . 193
7.1.3.428 acos_conjugate() . . . 193
7.1.3.429 REGISTER_FUNCTION() [28/36] . . . 193
7.1.3.430 atan_evalf() . . . 193
7.1.3.431 atan_eval() . . . 194
7.1.3.432 atan_deriv() . . . 194

7.1.3.433 atan_series() . . . 194
7.1.3.434 atan_conjugate() . . . 194
7.1.3.435 atan_info() . . . 195
7.1.3.436 REGISTER_FUNCTION() [29/36] . . . 195
7.1.3.437 atan2_evalf() . . . 195
7.1.3.438 atan2_eval() . . . 195
7.1.3.439 atan2_deriv() . . . 195
7.1.3.440 atan2_info() . . . 196
7.1.3.441 REGISTER_FUNCTION() [30/36] . . . 196
7.1.3.442 sinh_evalf() . . . 196
7.1.3.443 sinh_eval() . . . 196
7.1.3.444 sinh_deriv() . . . 196
7.1.3.445 sinh_real_part() . . . 197
7.1.3.446 sinh_imag_part() . . . 197
7.1.3.447 sinh_conjugate() . . . 197
7.1.3.448 REGISTER_FUNCTION() [31/36] . . . 197
7.1.3.449 cosh_evalf() . . . 197
7.1.3.450 cosh_eval() . . . 198
7.1.3.451 cosh_deriv() . . . 198
7.1.3.452 cosh_real_part() . . . 198
7.1.3.453 cosh_imag_part() . . . 198
7.1.3.454 cosh_conjugate() . . . 198
7.1.3.455 REGISTER_FUNCTION() [32/36] . . . 199
7.1.3.456 tanh_evalf() . . . 199
7.1.3.457 tanh_eval() . . . 199
7.1.3.458 tanh_deriv() . . . 199
7.1.3.459 tanh_series() . . . 199
7.1.3.460 tanh_real_part() . . . 200
7.1.3.461 tanh_imag_part() . . . 200
7.1.3.462 tanh_conjugate() . . . 200
7.1.3.463 REGISTER_FUNCTION() [33/36] . . . 200
7.1.3.464 asinh_evalf() . . . 200
7.1.3.465 asinh_eval() . . . 201
7.1.3.466 asinh_deriv() . . . 201
7.1.3.467 asinh_conjugate() . . . 201
7.1.3.468 REGISTER_FUNCTION() [34/36] . . . 201
7.1.3.469 acosh_evalf() . . . 201
7.1.3.470 acosh_eval() . . . 202
7.1.3.471 acosh_deriv() . . . 202
7.1.3.472 acosh_conjugate() . . . 202
7.1.3.473 REGISTER_FUNCTION() [35/36] . . . 202
7.1.3.474 atanh_evalf() . . . 202

7.1.3.475 atanh_eval() . . . 203
7.1.3.476 atanh_deriv() . . . 203
7.1.3.477 atanh_series() . . . 203
7.1.3.478 atanh_conjugate() . . . 203
7.1.3.479 REGISTER_FUNCTION() [36/36] . . . 203
7.1.3.480 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [12/34] . . . 204
7.1.3.481 subsvalue() . . . 204
7.1.3.482 adaptivesimpson() . . . 204
7.1.3.483 GINAC_BIND_UNARCHIVER() [22/50] . . . 204
7.1.3.484 GINAC_DECLARE_UNARCHIVER() [23/52] . . . 205
7.1.3.485 ifactor() . . . 205
7.1.3.486 is_discriminant_of_quadratic_number_field() . . . 205
7.1.3.487 kronecker_symbol() . . . 205
7.1.3.488 primitive_dirichlet_character() . . . 206
7.1.3.489 dirichlet_character() . . . 206
7.1.3.490 generalised_Bernoulli_number() . . . 206
7.1.3.491 Bernoulli_polynomial() . . . 207
7.1.3.492 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [13/34] . . . 207
7.1.3.493 GINAC_BIND_UNARCHIVER() [23/50] . . . 207
7.1.3.494 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [14/34] . . . 207
7.1.3.495 GINAC_BIND_UNARCHIVER() [24/50] . . . 207
7.1.3.496 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [15/34] . . . 208
7.1.3.497 GINAC_BIND_UNARCHIVER() [25/50] . . . 208
7.1.3.498 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [16/34] . . . 208
7.1.3.499 GINAC_BIND_UNARCHIVER() [26/50] . . . 208
7.1.3.500 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [17/34] . . . 208
7.1.3.501 GINAC_BIND_UNARCHIVER() [27/50] . . . 208
7.1.3.502 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [18/34] . . . 209
7.1.3.503 GINAC_BIND_UNARCHIVER() [28/50] . . . 209
7.1.3.504 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [19/34] . . . 209
7.1.3.505 GINAC_BIND_UNARCHIVER() [29/50] . . . 209
7.1.3.506 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [20/34] . . . 209
7.1.3.507 GINAC_BIND_UNARCHIVER() [30/50] . . . 209
7.1.3.508 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [21/34] . . . 210
7.1.3.509 GINAC_BIND_UNARCHIVER() [31/50] . . . 210
7.1.3.510 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [22/34] . . . 210
7.1.3.511 GINAC_BIND_UNARCHIVER() [32/50] . . . 210
7.1.3.512 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [23/34] . . . 210
7.1.3.513 GINAC_BIND_UNARCHIVER() [33/50] . . . 210
7.1.3.514 GINAC_DECLARE_UNARCHIVER() [24/52] . . . 211
7.1.3.515 GINAC_DECLARE_UNARCHIVER() [25/52] . . . 211
7.1.3.516 GINAC_DECLARE_UNARCHIVER() [26/52] . . . 211

7.1.3.517 GINAC_DECLARE_UNARCHIVER() [27/52] . . . 211
7.1.3.518 GINAC_DECLARE_UNARCHIVER() [28/52] . . . 211
7.1.3.519 GINAC_DECLARE_UNARCHIVER() [29/52] . . . 211
7.1.3.520 GINAC_DECLARE_UNARCHIVER() [30/52] . . . 211
7.1.3.521 GINAC_DECLARE_UNARCHIVER() [31/52] . . . 212
7.1.3.522 GINAC_DECLARE_UNARCHIVER() [32/52] . . . 212
7.1.3.523 GINAC_DECLARE_UNARCHIVER() [33/52] . . . 212
7.1.3.524 GINAC_DECLARE_UNARCHIVER() [34/52] . . . 212
7.1.3.525 GINAC_DECLARE_UNARCHIVER() [35/52] . . . 212
7.1.3.526 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [24/34] . . . 212
7.1.3.527 GINAC_BIND_UNARCHIVER() [34/50] . . . 213
7.1.3.528 lst_to_matrix() . . . 213
7.1.3.529 diag_matrix() [1/2] . . . 213
7.1.3.530 diag_matrix() [2/2] . . . 213
7.1.3.531 unit_matrix() [1/2] . . . 213
7.1.3.532 symbolic_matrix() [1/2] . . . 214
7.1.3.533 reduced_matrix() . . . 214
7.1.3.534 sub_matrix() . . . 214
7.1.3.535 GINAC_DECLARE_UNARCHIVER() [36/52] . . . 214
7.1.3.536 nops() [2/2] . . . 215
7.1.3.537 expand() [2/2] . . . 215
7.1.3.538 evalf() [2/2] . . . 215
7.1.3.539 rows() . . . 215
7.1.3.540 cols() . . . 215
7.1.3.541 transpose() . . . 215
7.1.3.542 determinant() . . . 216
7.1.3.543 trace() . . . 216
7.1.3.544 charpoly() . . . 216
7.1.3.545 inverse() [1/3] . . . 216
7.1.3.546 inverse() [2/3] . . . 216
7.1.3.547 rank() [1/2] . . . 217
7.1.3.548 rank() [2/2] . . . 217
7.1.3.549 unit_matrix() [2/2] . . . 217
7.1.3.550 symbolic_matrix() [2/2] . . . 217
7.1.3.551 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [25/34] . . . 217
7.1.3.552 tryfactsubs() . . . 218
7.1.3.553 algebraic_match_mul_with_mul() . . . 218
7.1.3.554 GINAC_BIND_UNARCHIVER() [35/50] . . . 218
7.1.3.555 GINAC_DECLARE_UNARCHIVER() [37/52] . . . 218
7.1.3.556 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [26/34] . . . 219
7.1.3.557 reeval_ncmul() . . . 219
7.1.3.558 hold_ncmul() . . . 219

7.1.3.559 GINAC_BIND_UNARCHIVER() [36/50] . . . 219
7.1.3.560 GINAC_DECLARE_UNARCHIVER() [38/52] . . . 219
7.1.3.561 get_first_symbol() . . . 220
7.1.3.562 add_symbol() . . . 220
7.1.3.563 collect_symbols() . . . 220
7.1.3.564 get_symbol_stats() . . . 221
7.1.3.565 lcmcoeff() . . . 221
7.1.3.566 lcm_of_coefficients_denominators() . . . 221
7.1.3.567 multiply_lcm() . . . 222
7.1.3.568 quo() . . . 222
7.1.3.569 rem() . . . 223
7.1.3.570 decomp_rational() . . . 223
7.1.3.571 prem() . . . 224
7.1.3.572 sprem() . . . 224
7.1.3.573 divide() . . . 225
7.1.3.574 divide_in_z() . . . 225
7.1.3.575 sr_gcd() . . . 226
7.1.3.576 interpolate() . . . 226
7.1.3.577 heur_gcd_z() . . . 227
7.1.3.578 heur_gcd() . . . 228
7.1.3.579 gcd_pf_pow() . . . 228
7.1.3.580 gcd_pf_mul() . . . 229
7.1.3.581 gcd() [1/2] . . . 229
7.1.3.582 gcd_pf_pow_pow() . . . 230
7.1.3.583 lcm() [1/2] . . . 230
7.1.3.584 sqrfree_yun() . . . 230
7.1.3.585 sqrfree() . . . 231
7.1.3.586 sqrfree_parfrac() . . . 232
7.1.3.587 replace_with_symbol() [1/2] . . . 232
7.1.3.588 replace_with_symbol() [2/2] . . . 233
7.1.3.589 frac_cancel() . . . 233
7.1.3.590 find_common_factor() . . . 233
7.1.3.591 collect_common_factors() . . . 234
7.1.3.592 resultant() . . . 234
7.1.3.593 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [27/34] . . . 234
7.1.3.594 make_real_float() . . . 235
7.1.3.595 read_real_float() . . . 235
7.1.3.596 GINAC_BIND_UNARCHIVER() [37/50] . . . 235
7.1.3.597 write_real_float() . . . 235
7.1.3.598 print_real_number() . . . 235
7.1.3.599 print_integer_csrc() . . . 236
7.1.3.600 print_real_csrc() . . . 236

7.1.3.601 coerce() . . . 236
7.1.3.602 coerce< int, cln::cl_I >() . . . 236
7.1.3.603 coerce< unsigned int, cln::cl_I >() . . . 237
7.1.3.604 print_real_cl_N() . . . 237
7.1.3.605 exp() . . . 237
7.1.3.606 log() . . . 238
7.1.3.607 sin() . . . 238
7.1.3.608 cos() . . . 239
7.1.3.609 tan() . . . 239
7.1.3.610 asin() . . . 239
7.1.3.611 acos() . . . 240
7.1.3.612 atan() [1/2] . . . 240
7.1.3.613 atan() [2/2] . . . 241
7.1.3.614 sinh() . . . 241
7.1.3.615 cosh() . . . 241
7.1.3.616 tanh() . . . 242
7.1.3.617 asinh() . . . 242
7.1.3.618 acosh() . . . 242
7.1.3.619 atanh() . . . 243
7.1.3.620 Li2_series() [2/2] . . . 243
7.1.3.621 Li2_projection() . . . 243
7.1.3.622 Li2_() . . . 243
7.1.3.623 Li2() . . . 244
7.1.3.624 zeta() [3/3] . . . 244
7.1.3.625 guess_precision() . . . 244
7.1.3.626 lgamma() [1/2] . . . 244
7.1.3.627 lgamma() [2/2] . . . 244
7.1.3.628 tgamma() [1/2] . . . 245
7.1.3.629 tgamma() [2/2] . . . 245
7.1.3.630 psi() [3/4] . . . 245
7.1.3.631 psi() [4/4] . . . 245
7.1.3.632 factorial() . . . 246
7.1.3.633 doublefactorial() . . . 246
7.1.3.634 binomial() . . . 247
7.1.3.635 bernoulli() . . . 247
7.1.3.636 fibonacci() . . . 248
7.1.3.637 abs() . . . 248
7.1.3.638 mod() . . . 248
7.1.3.639 smod() . . . 249
7.1.3.640 irem() [1/2] . . . 249
7.1.3.641 irem() [2/2] . . . 250
7.1.3.642 iquo() [1/2] . . . 250

7.1.3.643 iquo() [2/2] . . . 251
7.1.3.644 gcd() [2/2] . . . 251
7.1.3.645 lcm() [2/2] . . . 251
7.1.3.646 sqrt() [1/2] . . . 252
7.1.3.647 isqrt() . . . 252
7.1.3.648 PiEvalf() . . . 252
7.1.3.649 EulerEvalf() . . . 252
7.1.3.650 CatalanEvalf() . . . 253
7.1.3.651 operator<<() [6/16] . . . 253
7.1.3.652 GINAC_DECLARE_UNARCHIVER() [39/52] . . . 253
7.1.3.653 pow() [1/3] . . . 253
7.1.3.654 inverse() [3/3] . . . 254
7.1.3.655 step() . . . 254
7.1.3.656 csgn() . . . 254
7.1.3.657 is_zero() [2/2] . . . 254
7.1.3.658 is_positive() . . . 254
7.1.3.659 is_negative() . . . 254
7.1.3.660 is_integer() . . . 255
7.1.3.661 is_pos_integer() . . . 255
7.1.3.662 is_nonneg_integer() . . . 255
7.1.3.663 is_even() . . . 255
7.1.3.664 is_odd() . . . 255
7.1.3.665 is_prime() . . . 256
7.1.3.666 is_rational() . . . 256
7.1.3.667 is_real() . . . 256
7.1.3.668 is_cinteger() . . . 256
7.1.3.669 is_crational() . . . 256
7.1.3.670 to_int() . . . 257
7.1.3.671 to_long() . . . 257
7.1.3.672 to_double() . . . 257
7.1.3.673 real() . . . 257
7.1.3.674 imag() . . . 257
7.1.3.675 numer() [2/2] . . . 258
7.1.3.676 denom() [2/2] . . . 258
7.1.3.677 exadd() . . . 258
7.1.3.678 exmul() . . . 258
7.1.3.679 exminus() . . . 258
7.1.3.680 operator+() [1/4] . . . 259
7.1.3.681 operator-() [1/4] . . . 259
7.1.3.682 operator*() [1/2] . . . 259
7.1.3.683 operator/() [1/2] . . . 259
7.1.3.684 operator+() [2/4] . . . 259

7.1.3.685 operator-() [2/4] . . . 260
7.1.3.686 operator*() [2/2] . . . 260
7.1.3.687 operator/() [2/2] . . . 260
7.1.3.688 operator+=() [1/2] . . . 260
7.1.3.689 operator-=() [1/2] . . . 260
7.1.3.690 operator*=() [1/2] . . . 260
7.1.3.691 operator/=() [1/2] . . . 261
7.1.3.692 operator+=() [2/2] . . . 261
7.1.3.693 operator-=() [2/2] . . . 261
7.1.3.694 operator*=() [2/2] . . . 261
7.1.3.695 operator/=() [2/2] . . . 261
7.1.3.696 operator+() [3/4] . . . 261
7.1.3.697 operator-() [3/4] . . . 262
7.1.3.698 operator+() [4/4] . . . 262
7.1.3.699 operator-() [4/4] . . . 262
7.1.3.700 operator++() [1/4] . . . 262
7.1.3.701 operator--() [1/4] . . . 262
7.1.3.702 operator++() [2/4] . . . 263
7.1.3.703 operator--() [2/4] . . . 263
7.1.3.704 operator++() [3/4] . . . 263
7.1.3.705 operator--() [3/4] . . . 263
7.1.3.706 operator++() [4/4] . . . 264
7.1.3.707 operator--() [4/4] . . . 264
7.1.3.708 operator==() . . . 264
7.1.3.709 operator"!=() . . . 264
7.1.3.710 operator<() . . . 264
7.1.3.711 operator<=() . . . 265
7.1.3.712 operator>() . . . 265
7.1.3.713 operator>=() . . . 265
7.1.3.714 my_ios_index() . . . 265
7.1.3.715 my_ios_callback() . . . 265
7.1.3.716 get_print_context() . . . 266
7.1.3.717 set_print_context() . . . 266
7.1.3.718 get_print_options() . . . 266
7.1.3.719 set_print_options() . . . 266
7.1.3.720 operator<<() [7/16] . . . 266
7.1.3.721 operator>>() [3/3] . . . 267
7.1.3.722 dflt() . . . 267
7.1.3.723 latex() . . . 267
7.1.3.724 python() . . . 267
7.1.3.725 python_repr() . . . 267
7.1.3.726 tree() . . . 267

7.1.3.727 csrc() . . . 268
7.1.3.728 csrc_float() . . . 268
7.1.3.729 csrc_double() . . . 268
7.1.3.730 csrc_cl_N() . . . 268
7.1.3.731 index_dimensions() . . . 268
7.1.3.732 no_index_dimensions() . . . 269
7.1.3.733 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [28/34] . . . 269
7.1.3.734 print_sym_pow() . . . 269
7.1.3.735 GINAC_BIND_UNARCHIVER() [38/50] . . . 269
7.1.3.736 GINAC_DECLARE_UNARCHIVER() [40/52] . . . 269
7.1.3.737 pow() [2/3] . . . 270
7.1.3.738 pow() [3/3] . . . 270
7.1.3.739 sqrt() [2/2] . . . 270
7.1.3.740 is_a() [3/3] . . . 270
7.1.3.741 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [29/34] . . . 271
7.1.3.742 GINAC_BIND_UNARCHIVER() [39/50] . . . 271
7.1.3.743 GINAC_DECLARE_UNARCHIVER() [41/52] . . . 271
7.1.3.744 series_to_poly() . . . 271
7.1.3.745 is_terminating() . . . 272
7.1.3.746 make_return_type_t() . . . 272
7.1.3.747 set_print_func() [1/2] . . . 272
7.1.3.748 set_print_func() [2/2] . . . 272
7.1.3.749 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [30/34] . . . 272
7.1.3.750 GINAC_BIND_UNARCHIVER() [40/50] . . . 273
7.1.3.751 print_operator() . . . 273
7.1.3.752 GINAC_DECLARE_UNARCHIVER() [42/52] . . . 273
7.1.3.753 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [31/34] . . . 273
7.1.3.754 get_default_TeX_name() . . . 273
7.1.3.755 GINAC_BIND_UNARCHIVER() [41/50] . . . 274
7.1.3.756 GINAC_BIND_UNARCHIVER() [42/50] . . . 274
7.1.3.757 GINAC_BIND_UNARCHIVER() [43/50] . . . 274
7.1.3.758 GINAC_DECLARE_UNARCHIVER() [43/52] . . . 274
7.1.3.759 GINAC_DECLARE_UNARCHIVER() [44/52] . . . 274
7.1.3.760 GINAC_DECLARE_UNARCHIVER() [45/52] . . . 274
7.1.3.761 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [32/34] . . . 274
7.1.3.762 GINAC_BIND_UNARCHIVER() [44/50] . . . 275
7.1.3.763 index0() . . . 275
7.1.3.764 index1() . . . 275
7.1.3.765 index2() . . . 275
7.1.3.766 index3() . . . 275
7.1.3.767 not_symmetric() . . . 275
7.1.3.768 symmetric2() . . . 276

7.1.3.769 symmetric3() . . . 276
7.1.3.770 symmetric4() . . . 276
7.1.3.771 antisymmetric2() . . . 276
7.1.3.772 antisymmetric3() . . . 276
7.1.3.773 antisymmetric4() . . . 276
7.1.3.774 canonicalize() . . . 277
7.1.3.775 symm() . . . 277
7.1.3.776 symmetrize() [3/4] . . . 277
7.1.3.777 antisymmetrize() [3/4] . . . 277
7.1.3.778 symmetrize_cyclic() [3/4] . . . 278
7.1.3.779 GINAC_DECLARE_UNARCHIVER() [46/52] . . . 278
7.1.3.780 sy_none() [1/4] . . . 278
7.1.3.781 sy_none() [2/4] . . . 278
7.1.3.782 sy_none() [3/4] . . . 278
7.1.3.783 sy_none() [4/4] . . . 279
7.1.3.784 sy_symm() [1/4] . . . 279
7.1.3.785 sy_symm() [2/4] . . . 279
7.1.3.786 sy_symm() [3/4] . . . 279
7.1.3.787 sy_symm() [4/4] . . . 279
7.1.3.788 sy_anti() [1/4] . . . 280
7.1.3.789 sy_anti() [2/4] . . . 280
7.1.3.790 sy_anti() [3/4] . . . 280
7.1.3.791 sy_anti() [4/4] . . . 280
7.1.3.792 sy_cycl() [1/4] . . . 280
7.1.3.793 sy_cycl() [2/4] . . . 280
7.1.3.794 sy_cycl() [3/4] . . . 281
7.1.3.795 sy_cycl() [4/4] . . . 281
7.1.3.796 symmetrize() [4/4] . . . 281
7.1.3.797 antisymmetrize() [4/4] . . . 281
7.1.3.798 symmetrize_cyclic() [4/4] . . . 281
7.1.3.799 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [33/34] . . . 282
7.1.3.800 print_func< print_dflt >() [3/3] . . . 282
7.1.3.801 GINAC_BIND_UNARCHIVER() [45/50] . . . 282
7.1.3.802 GINAC_BIND_UNARCHIVER() [46/50] . . . 282
7.1.3.803 GINAC_BIND_UNARCHIVER() [47/50] . . . 282
7.1.3.804 GINAC_BIND_UNARCHIVER() [48/50] . . . 282
7.1.3.805 GINAC_BIND_UNARCHIVER() [49/50] . . . 283
7.1.3.806 delta_tensor() . . . 283
7.1.3.807 metric_tensor() . . . 283
7.1.3.808 lorentz_g() . . . 284
7.1.3.809 spinor_metric() . . . 284
7.1.3.810 epsilon_tensor() [1/2] . . . 285

7.1.3.811 epsilon_tensor() [2/2] . . . 285
7.1.3.812 lorentz_eps() . . . 286
7.1.3.813 GINAC_DECLARE_UNARCHIVER() [47/52] . . . 286
7.1.3.814 GINAC_DECLARE_UNARCHIVER() [48/52] . . . 286
7.1.3.815 GINAC_DECLARE_UNARCHIVER() [49/52] . . . 286
7.1.3.816 GINAC_DECLARE_UNARCHIVER() [50/52] . . . 287
7.1.3.817 GINAC_DECLARE_UNARCHIVER() [51/52] . . . 287
7.1.3.818 log2() . . . 287
7.1.3.819 multinomial_coefficient() . . . 287
7.1.3.820 rotate_left() . . . 287
7.1.3.821 compare_pointers() . . . 288
7.1.3.822 golden_ratio_hash() . . . 288
7.1.3.823 permutation_sign() [1/2] . . . 288
7.1.3.824 permutation_sign() [2/2] . . . 288
7.1.3.825 shaker_sort() . . . 289
7.1.3.826 cyclic_permutation() . . . 289
7.1.3.827 format_index_value() [1/2] . . . 289
7.1.3.828 format_index_value() [2/2] . . . 289
7.1.3.829 operator<<() [8/16] . . . 290
7.1.3.830 operator<<() [9/16] . . . 290
7.1.3.831 operator<<() [10/16] . . . 290
7.1.3.832 operator<<() [11/16] . . . 290
7.1.3.833 operator<<() [12/16] . . . 291
7.1.3.834 operator<<() [13/16] . . . 291
7.1.3.835 operator<<() [14/16] . . . 291
7.1.3.836 operator<<() [15/16] . . . 291
7.1.3.837 operator<<() [16/16] . . . 292
7.1.3.838 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [34/34] . . . 292
7.1.3.839 GINAC_BIND_UNARCHIVER() [50/50] . . . 292
7.1.3.840 haswild() . . . 292
7.1.3.841 GINAC_DECLARE_UNARCHIVER() [52/52] . . . 292
7.1.3.842 wild() . . . 293
7.1.4 Variable Documentation . . . 293
7.1.4.1 unarch_table_instance . . . 293
7.1.4.2 map_evalm . . . 293
7.1.4.3 map_eval_integ . . . 293
7.1.4.4 class_info< OPT >::first . . . 293
7.1.4.5 class_info< OPT >::parents_identified . . . 293
7.1.4.6 tensor . . . 293
7.1.4.7 Pi . . . 294
7.1.4.8 Euler . . . 294
7.1.4.9 Catalan . . . 294

7.1.4.10 crctab . . . 294
7.1.4.11 library_initializer . . . 295
7.1.4.12 _num0_bp . . . 295
7.1.4.13 idx . . . 295
7.1.4.14 force_include_tgamma . . . 295
7.1.4.15 force_include_zeta1 . . . 295
7.1.4.16 GINAC_BIND_UNARCHIVER . . . 295
7.1.4.17 I . . . 296
7.1.4.18 Digits . . . 296
7.1.4.19 next_print_context_id . . . 296
7.1.4.20 structure< T, CP >::reg_info . . . 296
7.1.4.21 version_major . . . 296
7.1.4.22 version_minor . . . 297
7.1.4.23 version_micro . . . 297
7.1.4.24 _num_120_p . . . 297
7.1.4.25 _ex_120 . . . 297
7.1.4.26 _num_60_p . . . 297
7.1.4.27 _ex_60 . . . 297
7.1.4.28 _num_48_p . . . 297
7.1.4.29 _ex_48 . . . 297
7.1.4.30 _num_30_p . . . 298
7.1.4.31 _ex_30 . . . 298
7.1.4.32 _num_25_p . . . 298
7.1.4.33 _ex_25 . . . 298
7.1.4.34 _num_24_p . . . 298
7.1.4.35 _ex_24 . . . 298
7.1.4.36 _num_20_p . . . 298
7.1.4.37 _ex_20 . . . 298
7.1.4.38 _num_18_p . . . 299
7.1.4.39 _ex_18 . . . 299
7.1.4.40 _num_15_p . . . 299
7.1.4.41 _ex_15 . . . 299
7.1.4.42 _num_12_p . . . 299
7.1.4.43 _ex_12 . . . 299
7.1.4.44 _num_11_p . . . 299
7.1.4.45 _ex_11 . . . 299
7.1.4.46 _num_10_p . . . 300
7.1.4.47 _ex_10 . . . 300
7.1.4.48 _num_9_p . . . 300
7.1.4.49 _ex_9 . . . 300
7.1.4.50 _num_8_p . . . 300
7.1.4.51 _ex_8 . . . 300

7.1.4.52 _num_7_p . . . 300
7.1.4.53 _ex_7 . . . 300
7.1.4.54 _num_6_p . . . 301
7.1.4.55 _ex_6 . . . 301
7.1.4.56 _num_5_p . . . 301
7.1.4.57 _ex_5 . . . 301
7.1.4.58 _num_4_p . . . 301
7.1.4.59 _ex_4 . . . 301
7.1.4.60 _num_3_p . . . 301
7.1.4.61 _ex_3 . . . 301
7.1.4.62 _num_2_p . . . 302
7.1.4.63 _ex_2 . . . 302
7.1.4.64 _num_1_p . . . 302
7.1.4.65 _ex_1 . . . 302
7.1.4.66 _num_1_2_p . . . 302
7.1.4.67 _ex_1_2 . . . 302
7.1.4.68 _num_1_3_p . . . 303
7.1.4.69 _ex_1_3 . . . 303
7.1.4.70 _num_1_4_p . . . 303
7.1.4.71 _ex_1_4 . . . 303
7.1.4.72 _num0_p . . . 303
7.1.4.73 _ex0 . . . 304
7.1.4.74 _num1_4_p . . . 304
7.1.4.75 _ex1_4 . . . 304
7.1.4.76 _num1_3_p . . . 304
7.1.4.77 _ex1_3 . . . 304
7.1.4.78 _num1_2_p . . . 305
7.1.4.79 _ex1_2 . . . 305
7.1.4.80 _num1_p . . . 305
7.1.4.81 _ex1 . . . 305
7.1.4.82 _num2_p . . . 306
7.1.4.83 _ex2 . . . 306
7.1.4.84 _num3_p . . . 306
7.1.4.85 _ex3 . . . 306
7.1.4.86 _num4_p . . . 306
7.1.4.87 _ex4 . . . 307
7.1.4.88 _num5_p . . . 307
7.1.4.89 _ex5 . . . 307
7.1.4.90 _num6_p . . . 307
7.1.4.91 _ex6 . . . 307
7.1.4.92 _num7_p . . . 307
7.1.4.93 _ex7 . . . 307

7.1.4.94 _num8_p . . . 308
7.1.4.95 _ex8 . . . 308
7.1.4.96 _num9_p . . . 308
7.1.4.97 _ex9 . . . 308
7.1.4.98 _num10_p . . . 308
7.1.4.99 _ex10 . . . 308
7.1.4.100 _num11_p . . . 308
7.1.4.101 _ex11 . . . 309
7.1.4.102 _num12_p . . . 309
7.1.4.103 _ex12 . . . 309
7.1.4.104 _num15_p . . . 309
7.1.4.105 _ex15 . . . 309
7.1.4.106 _num18_p . . . 309
7.1.4.107 _ex18 . . . 309
7.1.4.108 _num20_p . . . 310
7.1.4.109 _ex20 . . . 310
7.1.4.110 _num24_p . . . 310
7.1.4.111 _ex24 . . . 310
7.1.4.112 _num25_p . . . 310
7.1.4.113 _ex25 . . . 310
7.1.4.114 _num30_p . . . 310
7.1.4.115 _ex30 . . . 310
7.1.4.116 _num48_p . . . 311
7.1.4.117 _ex48 . . . 311
7.1.4.118 _num60_p . . . 311
7.1.4.119 _ex60 . . . 311
7.1.4.120 _num120_p . . . 311
7.1.4.121 _ex120 . . . 311
7.2 GiNaC::Gt_detail Namespace Reference . . . 311
7.2.1 Function Documentation . . . 312
7.2.1.1 integer_partition() . . . 312
7.2.1.2 deconcatenate_path() . . . 312
7.2.1.3 deconcatenate_path< ex >() . . . 312
7.2.1.4 deconcatenate_path< Gt::kernel >() . . . 313
7.2.1.5 operator<<() . . . 313
7.2.1.6 operator<() . . . 313
7.3 GiNaC::internal Namespace Reference . . . 313
7.4 std Namespace Reference . . . 313
7.4.1 Function Documentation . . . 314
7.4.1.1 swap() . . . 314

8 Class Documentation **315**

8.1 GiNaC::internal::_iter_rep Struct Reference . . . 315
8.1.1 Constructor & Destructor Documentation . . . 316
8.1.1.1 _iter_rep() . . . 316
8.1.2 Member Function Documentation . . . 316
8.1.2.1 operator==() . . . 316
8.1.2.2 operator"!=() . . . 316
8.1.3 Member Data Documentation . . . 316
8.1.3.1 e . . . 316
8.1.3.2 i . . . 317
8.1.3.3 i_end . . . 317
8.2 GiNaC::_numeric_digits Class Reference . . . 317
8.2.1 Detailed Description . . . 318
8.2.2 Constructor & Destructor Documentation . . . 318
8.2.2.1 _numeric_digits() . . . 318
8.2.3 Member Function Documentation . . . 318
8.2.3.1 operator=() . . . 318
8.2.3.2 operator long() . . . 318
8.2.3.3 print() . . . 318
8.2.3.4 add_callback() . . . 319
8.2.4 Member Data Documentation . . . 319
8.2.4.1 digits . . . 319
8.2.4.2 too_late . . . 319
8.2.4.3 callbacklist . . . 319
8.3 GiNaC::add Class Reference . . . 320
8.3.1 Detailed Description . . . 326
8.3.2 Constructor & Destructor Documentation . . . 326
8.3.2.1 add() [1/6] . . . 326
8.3.2.2 add() [2/6] . . . 327
8.3.2.3 add() [3/6] . . . 327
8.3.2.4 add() [4/6] . . . 327
8.3.2.5 add() [5/6] . . . 327
8.3.2.6 add() [6/6] . . . 327
8.3.3 Member Function Documentation . . . 327
8.3.3.1 precedence() . . . 327
8.3.3.2 info() . . . 328
8.3.3.3 is_polynomial() . . . 328
8.3.3.4 degree() . . . 328
8.3.3.5 ldegree() . . . 328
8.3.3.6 coeff() . . . 329
8.3.3.7 eval() . . . 329
8.3.3.8 evalm() . . . 329
8.3.3.9 series() . . . 330

8.3.3.10 normal() . . . 330
8.3.3.11 integer_content() . . . 330
8.3.3.12 smod() . . . 331
8.3.3.13 max_coefficient() . . . 331
8.3.3.14 conjugate() . . . 331
8.3.3.15 real_part() . . . 332
8.3.3.16 imag_part() . . . 332
8.3.3.17 get_free_indices() . . . 332
8.3.3.18 eval_ncmul() . . . 332
8.3.3.19 derivative() . . . 332
8.3.3.20 return_type() . . . 333
8.3.3.21 return_type_tinfo() . . . 333
8.3.3.22 thisexpairseq() [1/2] . . . 333
8.3.3.23 thisexpairseq() [2/2] . . . 333
8.3.3.24 split_ex_to_pair() . . . 334
8.3.3.25 combine_ex_with_coeff_to_pair() . . . 334
8.3.3.26 combine_pair_with_coeff_to_pair() . . . 334
8.3.3.27 recombine_pair_to_ex() . . . 334
8.3.3.28 expand() . . . 335
8.3.3.29 print_add() . . . 335
8.3.3.30 do_print() . . . 335
8.3.3.31 do_print_latex() . . . 335
8.3.3.32 do_print_csrc() . . . 336
8.3.3.33 do_print_python_repr() . . . 336
8.3.4 Friends And Related Symbol Documentation . . . 336
8.3.4.1 mul . . . 336
8.3.4.2 power . . . 336
8.4 GiNaC::archive Class Reference . . . 337
8.4.1 Detailed Description . . . 338
8.4.2 Member Typedef Documentation . . . 338
8.4.2.1 inv_at_cit . . . 338
8.4.3 Constructor & Destructor Documentation . . . 338
8.4.3.1 archive() [1/3] . . . 338
8.4.3.2 ∼archive() . . . 338
8.4.3.3 archive() [2/3] . . . 339
8.4.3.4 archive() [3/3] . . . 339
8.4.4 Member Function Documentation . . . 339
8.4.4.1 archive_ex() . . . 339
8.4.4.2 unarchive_ex() [1/3] . . . 340
8.4.4.3 unarchive_ex() [2/3] . . . 340
8.4.4.4 unarchive_ex() [3/3] . . . 340
8.4.4.5 num_expressions() . . . 341

8.4.4.6 get_top_node() . . . 341
8.4.4.7 clear() . . . 341
8.4.4.8 add_node() . . . 341
8.4.4.9 get_node() . . . 341
8.4.4.10 forget() . . . 342
8.4.4.11 printraw() . . . 342
8.4.4.12 atomize() . . . 342
8.4.4.13 unatomize() . . . 342
8.4.5 Friends And Related Symbol Documentation . . . 343
8.4.5.1 operator<< . . . 343
8.4.5.2 operator>> . . . 343
8.4.6 Member Data Documentation . . . 343
8.4.6.1 nodes . . . 343
8.4.6.2 exprs . . . 343
8.4.6.3 atoms . . . 344
8.4.6.4 inverse_atoms . . . 344
8.4.6.5 exprtable . . . 344
8.5 GiNaC::archive_node Class Reference . . . 344
8.5.1 Detailed Description . . . 347
8.5.2 Member Typedef Documentation . . . 347
8.5.2.1 propinfovector . . . 347
8.5.2.2 archive_node_cit . . . 347
8.5.3 Member Enumeration Documentation . . . 348
8.5.3.1 property_type . . . 348
8.5.4 Constructor & Destructor Documentation . . . 348
8.5.4.1 archive_node() [1/2] . . . 348
8.5.4.2 archive_node() [2/2] . . . 348
8.5.5 Member Function Documentation . . . 348
8.5.5.1 operator=() . . . 348
8.5.5.2 add_bool() . . . 349
8.5.5.3 add_unsigned() . . . 349
8.5.5.4 add_string() . . . 349
8.5.5.5 add_ex() . . . 349
8.5.5.6 find_bool() . . . 350
8.5.5.7 find_unsigned() . . . 350
8.5.5.8 find_string() . . . 350
8.5.5.9 find_first() . . . 351
8.5.5.10 find_last() . . . 351
8.5.5.11 find_property_range() . . . 351
8.5.5.12 find_ex() . . . 351
8.5.5.13 find_ex_by_loc() . . . 352
8.5.5.14 find_ex_node() . . . 352

8.5.5.15 get_properties() . . . 352
8.5.5.16 unarchive() . . . 352
8.5.5.17 has_same_ex_as() . . . 353
8.5.5.18 has_ex() . . . 353
8.5.5.19 get_ex() . . . 353
8.5.5.20 forget() . . . 353
8.5.5.21 printraw() . . . 353
8.5.6 Friends And Related Symbol Documentation . . . 354
8.5.6.1 operator<< . . . 354
8.5.6.2 operator>> . . . 354
8.5.7 Member Data Documentation . . . 354
8.5.7.1 a . . . 354
8.5.7.2 props . . . 354
8.5.7.3 has_expression . . . 355
8.5.7.4 e . . . 355
8.6 GiNaC::archive_node::archive_node_cit_range Struct Reference . . . 355
8.6.1 Member Data Documentation . . . 356
8.6.1.1 begin . . . 356
8.6.1.2 end . . . 357
8.7 GiNaC::archive::archived_ex Struct Reference . . . 357
8.7.1 Detailed Description . . . 357
8.7.2 Constructor & Destructor Documentation . . . 357
8.7.2.1 archived_ex() [1/2] . . . 357
8.7.2.2 archived_ex() [2/2] . . . 357
8.7.3 Member Data Documentation . . . 358
8.7.3.1 name . . . 358
8.7.3.2 root . . . 358
8.8 GiNaC::basic Class Reference . . . 358
8.8.1 Detailed Description . . . 363
8.8.2 Constructor & Destructor Documentation . . . 363
8.8.2.1 basic() [1/2] . . . 363
8.8.2.2 ~basic() . . . 363
8.8.2.3 basic() [2/2] . . . 363
8.8.3 Member Function Documentation . . . 364
8.8.3.1 operator=() . . . 364
8.8.3.2 duplicate() . . . 364
8.8.3.3 eval() . . . 364
8.8.3.4 evalf() . . . 364
8.8.3.5 evalm() . . . 365
8.8.3.6 eval_integ() . . . 365
8.8.3.7 eval_ncmul() . . . 365
8.8.3.8 eval_indexed() . . . 365

8.8.3.9 print() . . . 366
8.8.3.10 dbgprint() . . . 366
8.8.3.11 dbgprinttree() . . . 366
8.8.3.12 precedence() . . . 367
8.8.3.13 info() . . . 367
8.8.3.14 nops() . . . 367
8.8.3.15 op() . . . 368
8.8.3.16 operator[]() [1/4] . . . 368
8.8.3.17 operator[]() [2/4] . . . 368
8.8.3.18 let_op() . . . 368
8.8.3.19 operator[]() [3/4] . . . 369
8.8.3.20 operator[]() [4/4] . . . 369
8.8.3.21 has() . . . 369
8.8.3.22 match() . . . 369
8.8.3.23 match_same_type() . . . 370
8.8.3.24 subs() . . . 370
8.8.3.25 map() . . . 370
8.8.3.26 accept() . . . 371
8.8.3.27 is_polynomial() . . . 371
8.8.3.28 degree() . . . 371
8.8.3.29 ldegree() . . . 371
8.8.3.30 coeff() . . . 372
8.8.3.31 expand() . . . 372
8.8.3.32 collect() . . . 372
8.8.3.33 derivative() . . . 373
8.8.3.34 series() . . . 373
8.8.3.35 normal() . . . 374
8.8.3.36 to_rational() . . . 374
8.8.3.37 to_polynomial() . . . 374
8.8.3.38 integer_content() . . . 375
8.8.3.39 smod() . . . 375
8.8.3.40 max_coefficient() . . . 375
8.8.3.41 get_free_indices() . . . 376
8.8.3.42 add_indexed() . . . 376
8.8.3.43 scalar_mul_indexed() . . . 377
8.8.3.44 contract_with() . . . 377
8.8.3.45 return_type() . . . 378
8.8.3.46 return_type_tinfo() . . . 378
8.8.3.47 conjugate() . . . 378
8.8.3.48 real_part() . . . 378
8.8.3.49 imag_part() . . . 379
8.8.3.50 compare_same_type() . . . 379

8.8.3.51 is_equal_same_type() . . . 379
8.8.3.52 calchash() . . . 380
8.8.3.53 print_dispatch() [1/2] . . . 380
8.8.3.54 print_dispatch() [2/2] . . . 380
8.8.3.55 archive() . . . 381
8.8.3.56 read_archive() . . . 381
8.8.3.57 subs_one_level() . . . 382
8.8.3.58 diff() . . . 382
8.8.3.59 compare() . . . 382
8.8.3.60 is_equal() . . . 383
8.8.3.61 hold() . . . 383
8.8.3.62 gethash() . . . 384
8.8.3.63 setflag() . . . 384
8.8.3.64 clearflag() . . . 384
8.8.3.65 ensure_if_modifiable() . . . 385
8.8.3.66 do_print() . . . 385
8.8.3.67 do_print_tree() . . . 385
8.8.3.68 do_print_python_repr() . . . 385
8.8.4 Friends And Related Symbol Documentation . . . 386
8.8.4.1 ex . . . 386
8.8.5 Member Data Documentation . . . 387
8.8.5.1 flags . . . 387
8.8.5.2 hashvalue . . . 387
8.9 GiNaC::basic_log_kernel Class Reference . . . 388
8.9.1 Detailed Description . . . 393
8.9.2 Member Function Documentation . . . 393
8.9.2.1 series_coeff_impl() . . . 393
8.9.2.2 do_print() . . . 393
8.10 GiNaC::basic_multi_iterator< T > Class Template Reference . . . 394
8.10.1 Detailed Description . . . 395
8.10.2 Constructor & Destructor Documentation . . . 396
8.10.2.1 basic_multi_iterator() [1/3] . . . 396
8.10.2.2 basic_multi_iterator() [2/3] . . . 396
8.10.2.3 basic_multi_iterator() [3/3] . . . 396
8.10.2.4 ~basic_multi_iterator() . . . 397
8.10.3 Member Function Documentation . . . 397
8.10.3.1 size() . . . 397
8.10.3.2 overflow() . . . 397
8.10.3.3 get_vector() . . . 397
8.10.3.4 operator[]() [1/2] . . . 398
8.10.3.5 operator[]() [2/2] . . . 398
8.10.3.6 operator()() [1/2] . . . 398

8.10.3.7 operator()() [2/2] . . . 398
8.10.3.8 init() . . . 398
8.10.3.9 operator++() . . . 399
8.10.4 Friends And Related Symbol Documentation . . . 399
8.10.4.1 operator<< . . . 399
8.10.5 Member Data Documentation . . . 399
8.10.5.1 N . . . 399
8.10.5.2 B . . . 399
8.10.5.3 v . . . 400
8.10.5.4 flag_overflow . . . 400
8.11 GiNaC::basic_partition_generator Class Reference . . . 400
8.11.1 Detailed Description . . . 402
8.11.2 Constructor & Destructor Documentation . . . 402
8.11.2.1 basic_partition_generator() . . . 402
8.11.3 Member Data Documentation . . . 402
8.11.3.1 mpgen . . . 402
8.12 GiNaC::class_info< OPT > Class Template Reference . . . 402
8.12.1 Constructor & Destructor Documentation . . . 404
8.12.1.1 class_info() . . . 404
8.12.2 Member Function Documentation . . . 404
8.12.2.1 get_parent() . . . 404
8.12.2.2 find() . . . 404
8.12.2.3 dump_hierarchy() . . . 404
8.12.2.4 dump_tree() . . . 405
8.12.2.5 identify_parents() . . . 405
8.12.3 Member Data Documentation . . . 405
8.12.3.1 options . . . 405
8.12.3.2 first . . . 405
8.12.3.3 next . . . 405
8.12.3.4 parent . . . 405
8.12.3.5 parents_identified . . . 406
8.13 GiNaC::clifford Class Reference . . . 406
8.13.1 Detailed Description . . . 413
8.13.2 Constructor & Destructor Documentation . . . 413
8.13.2.1 clifford() [1/4] . . . 413
8.13.2.2 clifford() [2/4] . . . 414
8.13.2.3 clifford() [3/4] . . . 414
8.13.2.4 clifford() [4/4] . . . 414
8.13.3 Member Function Documentation . . . 414
8.13.3.1 precedence() . . . 414
8.13.3.2 archive() . . . 415
8.13.3.3 read_archive() . . . 415

8.13.3.4 eval_ncmul() . . . 415
8.13.3.5 match_same_type() . . . 416
8.13.3.6 thiscontainer() [1/2] . . . 416
8.13.3.7 thiscontainer() [2/2] . . . 416
8.13.3.8 return_type() . . . 416
8.13.3.9 return_type_tinfo() . . . 417
8.13.3.10 get_representation_label() . . . 417
8.13.3.11 get_metric() [1/2] . . . 417
8.13.3.12 get_metric() [2/2] . . . 417
8.13.3.13 same_metric() . . . 417
8.13.3.14 get_commutator_sign() . . . 417
8.13.3.15 nops() . . . 418
8.13.3.16 op() . . . 418
8.13.3.17 let_op() . . . 418
8.13.3.18 subs() . . . 418
8.13.3.19 do_print_dflt() . . . 419
8.13.3.20 do_print_latex() . . . 419
8.13.3.21 do_print_tree() . . . 419
8.13.4 Member Data Documentation . . . 419
8.13.4.1 representation_label . . . 419
8.13.4.2 metric . . . 419
8.13.4.3 commutator_sign . . . 420
8.14 GiNaC::cliffordunit Class Reference . . . 420
8.14.1 Detailed Description . . . 424
8.14.2 Member Function Documentation . . . 425
8.14.2.1 contract_with() . . . 425
8.14.2.2 do_print() . . . 425
8.14.2.3 do_print_latex() . . . 425
8.15 GiNaC::color Class Reference . . . 425
8.15.1 Detailed Description . . . 433
8.15.2 Constructor & Destructor Documentation . . . 433
8.15.2.1 color() [1/4] . . . 433
8.15.2.2 color() [2/4] . . . 433
8.15.2.3 color() [3/4] . . . 434
8.15.2.4 color() [4/4] . . . 434
8.15.3 Member Function Documentation . . . 434
8.15.3.1 archive() . . . 434
8.15.3.2 read_archive() . . . 434
8.15.3.3 eval_ncmul() . . . 435
8.15.3.4 match_same_type() . . . 435
8.15.3.5 thiscontainer() [1/2] . . . 435
8.15.3.6 thiscontainer() [2/2] . . . 436

8.15.3.7 return_type() . . . 436
8.15.3.8 return_type_tinfo() . . . 436
8.15.3.9 get_representation_label() . . . 436
8.15.4 Member Data Documentation . . . 436
8.15.4.1 representation_label . . . 436
8.16 GiNaC::compare_all_equal< T > Class Template Reference . . . 437
8.16.1 Detailed Description . . . 437
8.16.2 Constructor & Destructor Documentation . . . 437
8.16.2.1 ∼compare_all_equal() . . . 437
8.16.3 Member Function Documentation . . . 437
8.16.3.1 struct_is_equal() . . . 437
8.16.3.2 struct_compare() . . . 437
8.17 GiNaC::compare_bitwise< T > Class Template Reference . . . 438
8.17.1 Detailed Description . . . 438
8.17.2 Constructor & Destructor Documentation . . . 438
8.17.2.1 ∼compare_bitwise() . . . 438
8.17.3 Member Function Documentation . . . 438
8.17.3.1 struct_is_equal() . . . 438
8.17.3.2 struct_compare() . . . 438
8.18 GiNaC::compare_std_less< T > Class Template Reference . . . 439
8.18.1 Detailed Description . . . 439
8.18.2 Constructor & Destructor Documentation . . . 439
8.18.2.1 ∼compare_std_less() . . . 439
8.18.3 Member Function Documentation . . . 439
8.18.3.1 struct_is_equal() . . . 439
8.18.3.2 struct_compare() . . . 439
8.19 GiNaC::composition_generator Class Reference . . . 440
8.19.1 Detailed Description . . . 441
8.19.2 Constructor & Destructor Documentation . . . 441
8.19.2.1 composition_generator() . . . 441
8.19.3 Member Function Documentation . . . 441
8.19.3.1 get() . . . 441
8.19.3.2 next() . . . 441
8.19.4 Member Data Documentation . . . 441
8.19.4.1 cmgen . . . 441
8.19.4.2 atend . . . 441
8.19.4.3 trivial . . . 442
8.19.4.4 composition . . . 442
8.19.4.5 current_updated . . . 442
8.20 GiNaC::const_iterator Class Reference . . . 442
8.20.1 Member Typedef Documentation . . . 444
8.20.1.1 iterator_category . . . 444

8.20.1.2 value_type . . . 444
8.20.1.3 difference_type . . . 444
8.20.1.4 pointer . . . 444
8.20.1.5 reference . . . 445
8.20.2 Constructor & Destructor Documentation . . . 445
8.20.2.1 const_iterator() [1/2] . . . 445
8.20.2.2 const_iterator() [2/2] . . . 445
8.20.3 Member Function Documentation . . . 445
8.20.3.1 operator∗() . . . 445
8.20.3.2 operator->() . . . 445
8.20.3.3 operator[]() . . . 445
8.20.3.4 operator++() [1/2] . . . 445
8.20.3.5 operator++() [2/2] . . . 446
8.20.3.6 operator+=() . . . 446
8.20.3.7 operator+() . . . 446
8.20.3.8 operator--() [1/2] . . . 446
8.20.3.9 operator--() [2/2] . . . 446
8.20.3.10 operator-=() . . . 446
8.20.3.11 operator-() . . . 446
8.20.3.12 operator==() . . . 447
8.20.3.13 operator"!=() . . . 447
8.20.3.14 operator<() . . . 447
8.20.3.15 operator>() . . . 447
8.20.3.16 operator<=() . . . 447
8.20.3.17 operator>=() . . . 447
8.20.4 Friends And Related Symbol Documentation . . . 447
8.20.4.1 ex . . . 447
8.20.4.2 const_preorder_iterator . . . 448
8.20.4.3 const_postorder_iterator . . . 448
8.20.4.4 operator+ . . . 448
8.20.4.5 operator- . . . 448
8.20.5 Member Data Documentation . . . 448
8.20.5.1 e . . . 448
8.20.5.2 i . . . 448
8.21 GiNaC::const_postorder_iterator Class Reference . . . 449
8.21.1 Member Typedef Documentation . . . 449
8.21.1.1 iterator_category . . . 449
8.21.1.2 value_type . . . 449
8.21.1.3 difference_type . . . 449
8.21.1.4 pointer . . . 450
8.21.1.5 reference . . . 450
8.21.2 Constructor & Destructor Documentation . . . 450

8.21.2.1 const_postorder_iterator() [1/2] . . . 450
8.21.2.2 const_postorder_iterator() [2/2] . . . 450
8.21.3 Member Function Documentation . . . 450
8.21.3.1 operator∗() . . . 450
8.21.3.2 operator->() . . . 450
8.21.3.3 operator++() [1/2] . . . 450
8.21.3.4 operator++() [2/2] . . . 451
8.21.3.5 operator==() . . . 451
8.21.3.6 operator"!=() . . . 451
8.21.3.7 descend() . . . 451
8.21.3.8 increment() . . . 451
8.21.4 Member Data Documentation . . . 451
8.21.4.1 s . . . 451
8.22 GiNaC::const_preorder_iterator Class Reference . . . 452
8.22.1 Member Typedef Documentation . . . 452
8.22.1.1 iterator_category . . . 452
8.22.1.2 value_type . . . 452
8.22.1.3 difference_type . . . 452
8.22.1.4 pointer . . . 453
8.22.1.5 reference . . . 453
8.22.2 Constructor & Destructor Documentation . . . 453
8.22.2.1 const_preorder_iterator() [1/2] . . . 453
8.22.2.2 const_preorder_iterator() [2/2] . . . 453
8.22.3 Member Function Documentation . . . 453
8.22.3.1 operator∗() . . . 453
8.22.3.2 operator->() . . . 453
8.22.3.3 operator++() [1/2] . . . 453
8.22.3.4 operator++() [2/2] . . . 454
8.22.3.5 operator==() . . . 454
8.22.3.6 operator"!=() . . . 454
8.22.3.7 increment() . . . 454
8.22.4 Member Data Documentation . . . 454
8.22.4.1 s . . . 454
8.23 GiNaC::constant Class Reference . . . 455
8.23.1 Detailed Description . . . 460
8.23.2 Constructor & Destructor Documentation . . . 460
8.23.2.1 constant() [1/2] . . . 460
8.23.2.2 constant() [2/2] . . . 460
8.23.3 Member Function Documentation . . . 461
8.23.3.1 info() . . . 461
8.23.3.2 evalf() . . . 461
8.23.3.3 is_polynomial() . . . 461

8.23.3.4 conjugate() . . . 461
8.23.3.5 real_part() . . . 462
8.23.3.6 imag_part() . . . 462
8.23.3.7 archive() . . . 462
8.23.3.8 read_archive() . . . 462
8.23.3.9 derivative() . . . 463
8.23.3.10 is_equal_same_type() . . . 463
8.23.3.11 calchash() . . . 463
8.23.3.12 get_name() . . . 463
8.23.3.13 get_TeX_name() . . . 464
8.23.3.14 do_print() . . . 464
8.23.3.15 do_print_tree() . . . 464
8.23.3.16 do_print_latex() . . . 464
8.23.3.17 do_print_python_repr() . . . 464
8.23.4 Member Data Documentation . . . 465
8.23.4.1 name . . . 465
8.23.4.2 TeX_name . . . 465
8.23.4.3 ef . . . 465
8.23.4.4 number . . . 465
8.23.4.5 serial . . . 465
8.23.4.6 next_serial . . . 465
8.23.4.7 domain . . . 466
8.24 GiNaC::container< C > Class Template Reference . . . 466
8.24.1 Detailed Description . . . 471
8.24.2 Member Typedef Documentation . . . 472
8.24.2.1 STLT . . . 472
8.24.2.2 const_iterator . . . 472
8.24.2.3 const_reverse_iterator . . . 472
8.24.3 Constructor & Destructor Documentation . . . 472
8.24.3.1 container() [1/4] . . . 472
8.24.3.2 container() [2/4] . . . 472
8.24.3.3 container() [3/4] . . . 472
8.24.3.4 container() [4/4] . . . 472
8.24.4 Member Function Documentation . . . 473
8.24.4.1 get_default_flags() . . . 473
8.24.4.2 get_open_delim() . . . 473
8.24.4.3 get_close_delim() . . . 473
8.24.4.4 info() . . . 473
8.24.4.5 precedence() . . . 473
8.24.4.6 nops() . . . 474
8.24.4.7 op() . . . 474
8.24.4.8 let_op() . . . 474

8.24.4.9 subs() . . . 475
8.24.4.10 read_archive() . . . 475
8.24.4.11 archive() . . . 475
8.24.4.12 conjugate() . . . 476
8.24.4.13 real_part() . . . 476
8.24.4.14 imag_part() . . . 476
8.24.4.15 is_equal_same_type() . . . 476
8.24.4.16 thiscontainer() [1/2] . . . 477
8.24.4.17 thiscontainer() [2/2] . . . 477
8.24.4.18 printseq() . . . 477
8.24.4.19 sort_() [1/2] . . . 477
8.24.4.20 sort_() [2/2] . . . 478
8.24.4.21 unique_() [1/2] . . . 478
8.24.4.22 prepend() . . . 478
8.24.4.23 append() . . . 478
8.24.4.24 remove_first() . . . 478
8.24.4.25 remove_last() . . . 479
8.24.4.26 remove_all() . . . 479
8.24.4.27 sort() . . . 479
8.24.4.28 unique() . . . 479
8.24.4.29 begin() . . . 479
8.24.4.30 end() . . . 480
8.24.4.31 rbegin() . . . 480
8.24.4.32 rend() . . . 480
8.24.4.33 do_print() . . . 480
8.24.4.34 do_print_tree() . . . 480
8.24.4.35 do_print_python() . . . 480
8.24.4.36 do_print_python_repr() . . . 481
8.24.4.37 subschildren() . . . 481
8.24.4.38 unique_() [2/2] . . . 481
8.25 GiNaC::container_storage< C > Class Template Reference . . . 481
8.25.1 Detailed Description . . . 482
8.25.2 Member Typedef Documentation . . . 482
8.25.2.1 STLT . . . 482
8.25.3 Constructor & Destructor Documentation . . . 482
8.25.3.1 container_storage() [1/4] . . . 482
8.25.3.2 container_storage() [2/4] . . . 483
8.25.3.3 container_storage() [3/4] . . . 483
8.25.3.4 container_storage() [4/4] . . . 483
8.25.3.5 ~container_storage() . . . 483
8.25.4 Member Function Documentation . . . 483
8.25.4.1 reserve() [1/4] . . . 483

8.25.4.2 reserve() [2/4] . . . 483
8.25.4.3 reserve() [3/4] . . . 483
8.25.4.4 reserve() [4/4] . . . 484
8.25.5 Member Data Documentation . . . 484
8.25.5.1 seq . . . 484
8.26 GiNaC::composition_generator::coolmulti Struct Reference . . . 485
8.26.1 Constructor & Destructor Documentation . . . 485
8.26.1.1 coolmulti() . . . 485
8.26.1.2 ∼coolmulti() . . . 486
8.26.2 Member Function Documentation . . . 486
8.26.2.1 next_permutation() . . . 486
8.26.2.2 finished() . . . 486
8.26.3 Member Data Documentation . . . 486
8.26.3.1 head . . . 486
8.26.3.2 i . . . 486
8.26.3.3 after_i . . . 486
8.27 GiNaC::derivative_map_function Struct Reference . . . 487
8.27.1 Detailed Description . . . 488
8.27.2 Constructor & Destructor Documentation . . . 488
8.27.2.1 derivative_map_function() . . . 488
8.27.3 Member Function Documentation . . . 488
8.27.3.1 operator()() . . . 488
8.27.4 Member Data Documentation . . . 488
8.27.4.1 s . . . 488
8.28 GiNaC::determinant_algo Class Reference . . . 489
8.28.1 Detailed Description . . . 489
8.28.2 Member Enumeration Documentation . . . 489
8.28.2.1 anonymous enum . . . 489
8.29 GiNaC::diracgamma Class Reference . . . 490
8.29.1 Detailed Description . . . 495
8.29.2 Member Function Documentation . . . 495
8.29.2.1 contract_with() . . . 495
8.29.2.2 do_print() . . . 495
8.29.2.3 do_print_latex() . . . 495
8.30 GiNaC::diracgamma5 Class Reference . . . 496
8.30.1 Detailed Description . . . 500
8.30.2 Member Function Documentation . . . 500
8.30.2.1 conjugate() . . . 500
8.30.2.2 do_print() . . . 500
8.30.2.3 do_print_latex() . . . 500
8.31 GiNaC::diracgammaL Class Reference . . . 501
8.31.1 Detailed Description . . . 505

8.31.2 Member Function Documentation . . . 505
8.31.2.1 conjugate() . . . 505
8.31.2.2 do_print() . . . 505
8.31.2.3 do_print_latex() . . . 505
8.32 GiNaC::diracgammaR Class Reference . . . 506
8.32.1 Detailed Description . . . 510
8.32.2 Member Function Documentation . . . 510
8.32.2.1 conjugate() . . . 510
8.32.2.2 do_print() . . . 510
8.32.2.3 do_print_latex() . . . 510
8.33 GiNaC::diracone Class Reference . . . 511
8.33.1 Detailed Description . . . 515
8.33.2 Member Function Documentation . . . 515
8.33.2.1 do_print() . . . 515
8.33.2.2 do_print_latex() . . . 515
8.34 GiNaC::do_taylor Class Reference . . . 516
8.34.1 Detailed Description . . . 516
8.35 GiNaC::domain Class Reference . . . 516
8.35.1 Detailed Description . . . 516
8.35.2 Member Enumeration Documentation . . . 516
8.35.2.1 anonymous enum . . . 516
8.36 GiNaC::dunno Class Reference . . . 517
8.36.1 Detailed Description . . . 517
8.37 GiNaC::Ebar_kernel Class Reference . . . 517
8.37.1 Detailed Description . . . 522
8.37.2 Constructor & Destructor Documentation . . . 523
8.37.2.1 Ebar_kernel() . . . 523
8.37.3 Member Function Documentation . . . 523
8.37.3.1 nops() . . . 523
8.37.3.2 op() . . . 523
8.37.3.3 let_op() . . . 523
8.37.3.4 is_numeric() . . . 524
8.37.3.5 get_numerical_value() . . . 524
8.37.3.6 series_coeff_impl() . . . 524
8.37.3.7 do_print() . . . 524
8.37.4 Member Data Documentation . . . 525
8.37.4.1 n . . . 525
8.37.4.2 m . . . 525
8.37.4.3 x . . . 525
8.37.4.4 y . . . 525
8.38 GiNaC::Eisenstein_h_kernel Class Reference . . . 525
8.38.1 Detailed Description . . . 531

8.38.2 Constructor & Destructor Documentation . . . 532
8.38.2.1 Eisenstein_h_kernel() . . . 532
8.38.3 Member Function Documentation . . . 532
8.38.3.1 series() . . . 532
8.38.3.2 nops() . . . 532
8.38.3.3 op() . . . 532
8.38.3.4 let_op() . . . 533
8.38.3.5 is_numeric() . . . 533
8.38.3.6 Laurent_series() . . . 533
8.38.3.7 get_numerical_value() . . . 533
8.38.3.8 uses_Laurent_series() . . . 534
8.38.3.9 coefficient_a0() . . . 534
8.38.3.10 coefficient_an() . . . 534
8.38.3.11 q_expansion_modular_form() . . . 534
8.38.3.12 do_print() . . . 535
8.38.4 Member Data Documentation . . . 535
8.38.4.1 k . . . 535
8.38.4.2 N . . . 535
8.38.4.3 r . . . 535
8.38.4.4 s . . . 535
8.38.4.5 C_norm . . . 535
8.39 GiNaC::Eisenstein_kernel Class Reference . . . 536
8.39.1 Detailed Description . . . 542
8.39.2 Constructor & Destructor Documentation . . . 542
8.39.2.1 Eisenstein_kernel() . . . 542
8.39.3 Member Function Documentation . . . 542
8.39.3.1 series() . . . 542
8.39.3.2 nops() . . . 543
8.39.3.3 op() . . . 543
8.39.3.4 let_op() . . . 543
8.39.3.5 is_numeric() . . . 543
8.39.3.6 Laurent_series() . . . 543
8.39.3.7 get_numerical_value() . . . 544
8.39.3.8 uses_Laurent_series() . . . 544
8.39.3.9 q_expansion_modular_form() . . . 544
8.39.3.10 do_print() . . . 544
8.39.4 Member Data Documentation . . . 544
8.39.4.1 k . . . 544
8.39.4.2 N . . . 545
8.39.4.3 a . . . 545
8.39.4.4 b . . . 545
8.39.4.5 K . . . 545

8.39.4.6 C_norm . . . 545
8.40 GiNaC::composition_generator::coolmulti::element Struct Reference . . . 545
8.40.1 Constructor & Destructor Documentation . . . 546
8.40.1.1 element() . . . 546
8.40.1.2 ∼element() . . . 546
8.40.2 Member Data Documentation . . . 546
8.40.2.1 value . . . 546
8.40.2.2 next . . . 546
8.41 GiNaC::ELi_kernel Class Reference . . . 547
8.41.1 Detailed Description . . . 552
8.41.2 Constructor & Destructor Documentation . . . 553
8.41.2.1 ELi_kernel() . . . 553
8.41.3 Member Function Documentation . . . 553
8.41.3.1 nops() . . . 553
8.41.3.2 op() . . . 553
8.41.3.3 let_op() . . . 553
8.41.3.4 is_numeric() . . . 554
8.41.3.5 get_numerical_value() . . . 554
8.41.3.6 series_coeff_impl() . . . 554
8.41.3.7 do_print() . . . 554
8.41.4 Member Data Documentation . . . 554
8.41.4.1 n . . . 554
8.41.4.2 m . . . 555
8.41.4.3 x . . . 555
8.41.4.4 y . . . 555
8.42 std::equal_to< GiNaC::ex > Struct Reference . . . 555
8.42.1 Detailed Description . . . 555
8.42.2 Member Function Documentation . . . 555
8.42.2.1 operator()() . . . 555
8.43 GiNaC::error_and_integral Struct Reference . . . 556
8.43.1 Constructor & Destructor Documentation . . . 557
8.43.1.1 error_and_integral() . . . 557
8.43.2 Member Data Documentation . . . 557
8.43.2.1 error . . . 557
8.43.2.2 integral . . . 557
8.44 GiNaC::error_and_integral_is_less Struct Reference . . . 557
8.44.1 Member Function Documentation . . . 557
8.44.1.1 operator()() . . . 557
8.45 GiNaC::eval_integ_map_function Struct Reference . . . 558
8.45.1 Detailed Description . . . 559
8.45.2 Member Function Documentation . . . 559
8.45.2.1 operator()() . . . 559

8.46 GiNaC::evalf_map_function Struct Reference . . . 559
8.46.1 Detailed Description . . . 560
8.46.2 Member Function Documentation . . . 560
8.46.2.1 operator()() . . . 560
8.47 GiNaC::evalm_map_function Struct Reference . . . 561
8.47.1 Detailed Description . . . 562
8.47.2 Member Function Documentation . . . 562
8.47.2.1 operator()() . . . 562
8.48 GiNaC::ex Class Reference . . . 562
8.48.1 Detailed Description . . . 566
8.48.2 Constructor & Destructor Documentation . . . 566
8.48.2.1 ex() [1/10] . . . 566
8.48.2.2 ex() [2/10] . . . 567
8.48.2.3 ex() [3/10] . . . 567
8.48.2.4 ex() [4/10] . . . 567
8.48.2.5 ex() [5/10] . . . 567
8.48.2.6 ex() [6/10] . . . 567
8.48.2.7 ex() [7/10] . . . 567
8.48.2.8 ex() [8/10] . . . 567
8.48.2.9 ex() [9/10] . . . 568
8.48.2.10 ex() [10/10] . . . 568
8.48.3 Member Function Documentation . . . 568
8.48.3.1 swap() . . . 568
8.48.3.2 begin() . . . 568
8.48.3.3 end() . . . 568
8.48.3.4 preorder_begin() . . . 569
8.48.3.5 preorder_end() . . . 569
8.48.3.6 postorder_begin() . . . 569
8.48.3.7 postorder_end() . . . 569
8.48.3.8 eval() . . . 569
8.48.3.9 evalf() . . . 569
8.48.3.10 evalm() . . . 570
8.48.3.11 eval_ncmul() . . . 570
8.48.3.12 eval_integ() . . . 570
8.48.3.13 print() . . . 570
8.48.3.14 dbgprint() . . . 571
8.48.3.15 dbgprinttree() . . . 571
8.48.3.16 info() . . . 571
8.48.3.17 nops() . . . 572
8.48.3.18 op() . . . 572
8.48.3.19 operator[]() [1/4] . . . 572
8.48.3.20 operator[]() [2/4] . . . 573

8.48.3.21 let_op() . . . 573
8.48.3.22 operator[]() [3/4] . . . 573
8.48.3.23 operator[]() [4/4] . . . 573
8.48.3.24 lhs() . . . 573
8.48.3.25 rhs() . . . 573
8.48.3.26 conjugate() . . . 574
8.48.3.27 real_part() . . . 574
8.48.3.28 imag_part() . . . 574
8.48.3.29 has() . . . 574
8.48.3.30 find() . . . 575
8.48.3.31 match() [1/2] . . . 575
8.48.3.32 match() [2/2] . . . 575
8.48.3.33 subs() [1/3] . . . 575
8.48.3.34 subs() [2/3] . . . 576
8.48.3.35 subs() [3/3] . . . 576
8.48.3.36 map() [1/2] . . . 576
8.48.3.37 map() [2/2] . . . 576
8.48.3.38 accept() . . . 576
8.48.3.39 traverse_preorder() . . . 577
8.48.3.40 traverse_postorder() . . . 577
8.48.3.41 traverse() . . . 577
8.48.3.42 is_polynomial() . . . 577
8.48.3.43 degree() . . . 577
8.48.3.44 ldegree() . . . 578
8.48.3.45 coeff() . . . 578
8.48.3.46 lcoeff() . . . 578
8.48.3.47 tcoeff() . . . 578
8.48.3.48 expand() . . . 579
8.48.3.49 collect() . . . 579
8.48.3.50 diff() . . . 579
8.48.3.51 series() . . . 580
8.48.3.52 normal() . . . 580
8.48.3.53 to_rational() . . . 581
8.48.3.54 to_polynomial() . . . 581
8.48.3.55 numer() . . . 581
8.48.3.56 denom() . . . 582
8.48.3.57 numer_denom() . . . 582
8.48.3.58 unit() . . . 583
8.48.3.59 content() . . . 583
8.48.3.60 integer_content() . . . 584
8.48.3.61 primpart() [1/2] . . . 584
8.48.3.62 primpart() [2/2] . . . 585

8.48.3.63 unitcontprim() . . . 585
8.48.3.64 smod() . . . 586
8.48.3.65 max_coefficient() . . . 586
8.48.3.66 get_free_indices() . . . 586
8.48.3.67 simplify_indexed() [1/2] . . . 587
8.48.3.68 simplify_indexed() [2/2] . . . 587
8.48.3.69 compare() . . . 587
8.48.3.70 is_equal() . . . 588
8.48.3.71 is_zero() . . . 588
8.48.3.72 is_zero_matrix() . . . 589
8.48.3.73 symmetrize() [1/2] . . . 589
8.48.3.74 symmetrize() [2/2] . . . 589
8.48.3.75 antisymmetrize() [1/2] . . . 589
8.48.3.76 antisymmetrize() [2/2] . . . 589
8.48.3.77 symmetrize_cyclic() [1/2] . . . 590
8.48.3.78 symmetrize_cyclic() [2/2] . . . 590
8.48.3.79 return_type() . . . 590
8.48.3.80 return_type_tinfo() . . . 590
8.48.3.81 gethash() . . . 590
8.48.3.82 construct_from_basic() . . . 591
8.48.3.83 construct_from_int() . . . 591
8.48.3.84 construct_from_uint() . . . 591
8.48.3.85 construct_from_long() . . . 591
8.48.3.86 construct_from_ulong() . . . 592
8.48.3.87 construct_from_longlong() . . . 592
8.48.3.88 construct_from_ulonglong() . . . 592
8.48.3.89 construct_from_double() . . . 592
8.48.3.90 construct_from_string_and_lst() . . . 592
8.48.3.91 makewriteable() . . . 593
8.48.3.92 share() . . . 593
8.48.4 Friends And Related Symbol Documentation . . . 593
8.48.4.1 archive_node . . . 593
8.48.4.2 are_ex_trivially_equal . . . 593
8.48.4.3 ex_to . . . 594
8.48.4.4 is_a . . . 594
8.48.4.5 is_exactly_a . . . 594
8.48.5 Member Data Documentation . . . 595
8.48.5.1 bp . . . 595
8.49 GiNaC::ex_base_is_less Struct Reference . . . 595
8.49.1 Member Function Documentation . . . 595
8.49.1.1 operator()() . . . 595
8.50 GiNaC::ex_is_equal Struct Reference . . . 595

8.50.1 Member Function Documentation . . . 596
8.50.1.1 operator()() . . . 596
8.51 GiNaC::ex_is_less Struct Reference . . . 596
8.51.1 Member Function Documentation . . . 596
8.51.1.1 operator()() . . . 596
8.52 GiNaC::ex_swap Struct Reference . . . 596
8.52.1 Member Function Documentation . . . 597
8.52.1.1 operator()() . . . 597
8.53 GiNaC::expair Class Reference . . . 597
8.53.1 Detailed Description . . . 599
8.53.2 Constructor & Destructor Documentation . . . 599
8.53.2.1 expair() [1/2] . . . 599
8.53.2.2 expair() [2/2] . . . 599
8.53.3 Member Function Documentation . . . 600
8.53.3.1 is_equal() . . . 600
8.53.3.2 is_less() . . . 600
8.53.3.3 compare() . . . 600
8.53.3.4 print() . . . 600
8.53.3.5 is_canonical_numeric() . . . 600
8.53.3.6 swap() . . . 601
8.53.3.7 conjugate() . . . 601
8.53.4 Member Data Documentation . . . 601
8.53.4.1 rest . . . 601
8.53.4.2 coeff . . . 601
8.54 GiNaC::expair_is_less Struct Reference . . . 602
8.54.1 Detailed Description . . . 602
8.54.2 Member Function Documentation . . . 602
8.54.2.1 operator()() . . . 602
8.55 GiNaC::expair_rest_is_less Struct Reference . . . 602
8.55.1 Detailed Description . . . 602
8.55.2 Member Function Documentation . . . 603
8.55.2.1 operator()() . . . 603
8.56 GiNaC::expair_swap Struct Reference . . . 603
8.56.1 Member Function Documentation . . . 603
8.56.1.1 operator()() . . . 603
8.57 GiNaC::expairseq Class Reference . . . 604
8.57.1 Detailed Description . . . 609
8.57.2 Constructor & Destructor Documentation . . . 609
8.57.2.1 expairseq() [1/4] . . . 609
8.57.2.2 expairseq() [2/4] . . . 610
8.57.2.3 expairseq() [3/4] . . . 610
8.57.2.4 expairseq() [4/4] . . . 610

8.57.3 Member Function Documentation . . . 610
8.57.3.1 precedence() . . . 610
8.57.3.2 info() . . . 610
8.57.3.3 nops() . . . 611
8.57.3.4 op() . . . 611
8.57.3.5 map() . . . 611
8.57.3.6 eval() . . . 611
8.57.3.7 to_rational() . . . 612
8.57.3.8 to_polynomial() . . . 612
8.57.3.9 match() . . . 612
8.57.3.10 subs() . . . 612
8.57.3.11 conjugate() . . . 613
8.57.3.12 archive() . . . 613
8.57.3.13 read_archive() . . . 613
8.57.3.14 is_equal_same_type() . . . 614
8.57.3.15 return_type() . . . 614
8.57.3.16 calchash() . . . 614
8.57.3.17 expand() . . . 614
8.57.3.18 thisexpairseq() [1/2] . . . 615
8.57.3.19 thisexpairseq() [2/2] . . . 615
8.57.3.20 printseq() . . . 615
8.57.3.21 printpair() . . . 615
8.57.3.22 split_ex_to_pair() . . . 616
8.57.3.23 combine_ex_with_coeff_to_pair() . . . 616
8.57.3.24 combine_pair_with_coeff_to_pair() . . . 616
8.57.3.25 recombine_pair_to_ex() . . . 616
8.57.3.26 expair_needs_further_processing() . . . 617
8.57.3.27 default_overall_coeff() . . . 617
8.57.3.28 combine_overall_coeff() [1/2] . . . 617
8.57.3.29 combine_overall_coeff() [2/2] . . . 617
8.57.3.30 can_make_flat() . . . 617
8.57.3.31 do_print() . . . 618
8.57.3.32 do_print_tree() . . . 618
8.57.3.33 construct_from_2_ex() . . . 618
8.57.3.34 construct_from_2_expairseq() . . . 618
8.57.3.35 construct_from_expairseq_ex() . . . 618
8.57.3.36 construct_from_exvector() . . . 619
8.57.3.37 construct_from_epvector() [1/2] . . . 619
8.57.3.38 construct_from_epvector() [2/2] . . . 619
8.57.3.39 make_flat() [1/2] . . . 619
8.57.3.40 make_flat() [2/2] . . . 619
8.57.3.41 canonicalize() . . . 620

8.57.3.42 combine_same_terms_sorted_seq() . . . 620
8.57.3.43 is_canonical() . . . 620
8.57.3.44 expandchildren() . . . 620
8.57.3.45 evalchildren() . . . 621
8.57.3.46 subschildren() . . . 621
8.57.4 Member Data Documentation . . . 621
8.57.4.1 seq . . . 621
8.57.4.2 overall_coeff . . . 622
8.58 GiNaC::expand_map_function Struct Reference . . . 622
8.58.1 Detailed Description . . . 623
8.58.2 Constructor & Destructor Documentation . . . 623
8.58.2.1 expand_map_function() . . . 623
8.58.3 Member Function Documentation . . . 624
8.58.3.1 operator()() . . . 624
8.58.4 Member Data Documentation . . . 624
8.58.4.1 options . . . 624
8.59 GiNaC::expand_options Class Reference . . . 624
8.59.1 Detailed Description . . . 624
8.59.2 Member Enumeration Documentation . . . 625
8.59.2.1 anonymous enum . . . 625
8.60 GiNaC::factor_options Class Reference . . . 625
8.60.1 Detailed Description . . . 625
8.60.2 Member Enumeration Documentation . . . 625
8.60.2.1 anonymous enum . . . 625
8.61 GiNaC::fail Class Reference . . . 626
8.61.1 Member Function Documentation . . . 630
8.61.1.1 return_type() . . . 630
8.61.1.2 do_print() . . . 630
8.62 GiNaC::fderivative Class Reference . . . 630
8.62.1 Detailed Description . . . 638
8.62.2 Constructor & Destructor Documentation . . . 638
8.62.2.1 fderivative() [1/3] . . . 638
8.62.2.2 fderivative() [2/3] . . . 639
8.62.2.3 fderivative() [3/3] . . . 639
8.62.3 Member Function Documentation . . . 639
8.62.3.1 print() . . . 639
8.62.3.2 eval() . . . 640
8.62.3.3 series() . . . 640
8.62.3.4 thiscontainer() [1/2] . . . 640
8.62.3.5 thiscontainer() [2/2] . . . 640
8.62.3.6 archive() . . . 641
8.62.3.7 read_archive() . . . 641

8.62.3.8 derivative() . . . 641
8.62.3.9 is_equal_same_type() . . . 642
8.62.3.10 match_same_type() . . . 642
8.62.3.11 derivatives() . . . 642
8.62.3.12 do_print() . . . 643
8.62.3.13 do_print_latex() . . . 643
8.62.3.14 do_print_csrc() . . . 643
8.62.3.15 do_print_tree() . . . 643
8.62.4 Member Data Documentation . . . 644
8.62.4.1 parameter_set . . . 644
8.63 GiNaC::function Class Reference . . . 644
8.63.1 Detailed Description . . . 651
8.63.2 Constructor & Destructor Documentation . . . 651
8.63.2.1 function() [1/18] . . . 651
8.63.2.2 function() [2/18] . . . 651
8.63.2.3 function() [3/18] . . . 652
8.63.2.4 function() [4/18] . . . 652
8.63.2.5 function() [5/18] . . . 652
8.63.2.6 function() [6/18] . . . 652
8.63.2.7 function() [7/18] . . . 652
8.63.2.8 function() [8/18] . . . 653
8.63.2.9 function() [9/18] . . . 653
8.63.2.10 function() [10/18] . . . 653
8.63.2.11 function() [11/18] . . . 653
8.63.2.12 function() [12/18] . . . 654
8.63.2.13 function() [13/18] . . . 654
8.63.2.14 function() [14/18] . . . 654
8.63.2.15 function() [15/18] . . . 655
8.63.2.16 function() [16/18] . . . 655
8.63.2.17 function() [17/18] . . . 655
8.63.2.18 function() [18/18] . . . 656
8.63.3 Member Function Documentation . . . 656
8.63.3.1 print() . . . 656
8.63.3.2 precedence() . . . 656
8.63.3.3 expand() . . . 657
8.63.3.4 eval() . . . 657
8.63.3.5 evalf() . . . 657
8.63.3.6 eval_ncmul() . . . 657
8.63.3.7 calchash() . . . 658
8.63.3.8 series() . . . 658
8.63.3.9 thiscontainer() [1/2] . . . 658
8.63.3.10 thiscontainer() [2/2] . . . 659

8.63.3.11 conjugate() . . . 659
8.63.3.12 real_part() . . . 659
8.63.3.13 imag_part() . . . 659
8.63.3.14 archive() . . . 660
8.63.3.15 read_archive() . . . 660
8.63.3.16 info() . . . 660
8.63.3.17 derivative() . . . 660
8.63.3.18 is_equal_same_type() . . . 661
8.63.3.19 match_same_type() . . . 661
8.63.3.20 return_type() . . . 661
8.63.3.21 return_type_tinfo() . . . 661
8.63.3.22 pderivative() . . . 662
8.63.3.23 expl_derivative() . . . 662
8.63.3.24 registered_functions() . . . 662
8.63.3.25 lookup_remember_table() . . . 662
8.63.3.26 store_remember_table() . . . 662
8.63.3.27 power() . . . 663
8.63.3.28 register_new() . . . 663
8.63.3.29 find_function() . . . 663
8.63.3.30 get_registered_functions() . . . 663
8.63.3.31 get_serial() . . . 663
8.63.3.32 get_name() . . . 663
8.63.4 Friends And Related Symbol Documentation . . . 664
8.63.4.1 remember_table_entry . . . 664
8.63.5 Member Data Documentation . . . 664
8.63.5.1 current_serial . . . 664
8.63.5.2 serial . . . 664
8.64 GiNaC::function_options Class Reference . . . 665
8.64.1 Constructor & Destructor Documentation . . . 671
8.64.1.1 function_options() [1/3] . . . 671
8.64.1.2 function_options() [2/3] . . . 671
8.64.1.3 function_options() [3/3] . . . 671
8.64.1.4 ~function_options() . . . 672
8.64.2 Member Function Documentation . . . 672
8.64.2.1 initialize() . . . 672
8.64.2.2 dummy() . . . 672
8.64.2.3 set_name() . . . 672
8.64.2.4 latex_name() . . . 672
8.64.2.5 eval_func() [1/15] . . . 672
8.64.2.6 eval_func() [2/15] . . . 673
8.64.2.7 eval_func() [3/15] . . . 673
8.64.2.8 eval_func() [4/15] . . . 673

8.64.2.9 eval_func() [5/15] . . . 673
8.64.2.10 eval_func() [6/15] . . . 673
8.64.2.11 eval_func() [7/15] . . . 673
8.64.2.12 eval_func() [8/15] . . . 673
8.64.2.13 eval_func() [9/15] . . . 674
8.64.2.14 eval_func() [10/15] . . . 674
8.64.2.15 eval_func() [11/15] . . . 674
8.64.2.16 eval_func() [12/15] . . . 674
8.64.2.17 eval_func() [13/15] . . . 674
8.64.2.18 eval_func() [14/15] . . . 674
8.64.2.19 evalf_func() [1/15] . . . 674
8.64.2.20 evalf_func() [2/15] . . . 675
8.64.2.21 evalf_func() [3/15] . . . 675
8.64.2.22 evalf_func() [4/15] . . . 675
8.64.2.23 evalf_func() [5/15] . . . 675
8.64.2.24 evalf_func() [6/15] . . . 675
8.64.2.25 evalf_func() [7/15] . . . 675
8.64.2.26 evalf_func() [8/15] . . . 675
8.64.2.27 evalf_func() [9/15] . . . 676
8.64.2.28 evalf_func() [10/15] . . . 676
8.64.2.29 evalf_func() [11/15] . . . 676
8.64.2.30 evalf_func() [12/15] . . . 676
8.64.2.31 evalf_func() [13/15] . . . 676
8.64.2.32 evalf_func() [14/15] . . . 676
8.64.2.33 conjugate_func() [1/15] . . . 676
8.64.2.34 conjugate_func() [2/15] . . . 677
8.64.2.35 conjugate_func() [3/15] . . . 677
8.64.2.36 conjugate_func() [4/15] . . . 677
8.64.2.37 conjugate_func() [5/15] . . . 677
8.64.2.38 conjugate_func() [6/15] . . . 677
8.64.2.39 conjugate_func() [7/15] . . . 677
8.64.2.40 conjugate_func() [8/15] . . . 677
8.64.2.41 conjugate_func() [9/15] . . . 678
8.64.2.42 conjugate_func() [10/15] . . . 678
8.64.2.43 conjugate_func() [11/15] . . . 678
8.64.2.44 conjugate_func() [12/15] . . . 678
8.64.2.45 conjugate_func() [13/15] . . . 678
8.64.2.46 conjugate_func() [14/15] . . . 678
8.64.2.47 real_part_func() [1/15] . . . 678
8.64.2.48 real_part_func() [2/15] . . . 679
8.64.2.49 real_part_func() [3/15] . . . 679
8.64.2.50 real_part_func() [4/15] . . . 679

8.64.2.51 real_part_func() [5/15] . . . 679
8.64.2.52 real_part_func() [6/15] . . . 679
8.64.2.53 real_part_func() [7/15] . . . 679
8.64.2.54 real_part_func() [8/15] . . . 679
8.64.2.55 real_part_func() [9/15] . . . 680
8.64.2.56 real_part_func() [10/15] . . . 680
8.64.2.57 real_part_func() [11/15] . . . 680
8.64.2.58 real_part_func() [12/15] . . . 680
8.64.2.59 real_part_func() [13/15] . . . 680
8.64.2.60 real_part_func() [14/15] . . . 680
8.64.2.61 imag_part_func() [1/15] . . . 680
8.64.2.62 imag_part_func() [2/15] . . . 681
8.64.2.63 imag_part_func() [3/15] . . . 681
8.64.2.64 imag_part_func() [4/15] . . . 681
8.64.2.65 imag_part_func() [5/15] . . . 681
8.64.2.66 imag_part_func() [6/15] . . . 681
8.64.2.67 imag_part_func() [7/15] . . . 681
8.64.2.68 imag_part_func() [8/15] . . . 681
8.64.2.69 imag_part_func() [9/15] . . . 682
8.64.2.70 imag_part_func() [10/15] . . . 682
8.64.2.71 imag_part_func() [11/15] . . . 682
8.64.2.72 imag_part_func() [12/15] . . . 682
8.64.2.73 imag_part_func() [13/15] . . . 682
8.64.2.74 imag_part_func() [14/15] . . . 682
8.64.2.75 expand_func() [1/15] . . . 682
8.64.2.76 expand_func() [2/15] . . . 683
8.64.2.77 expand_func() [3/15] . . . 683
8.64.2.78 expand_func() [4/15] . . . 683
8.64.2.79 expand_func() [5/15] . . . 683
8.64.2.80 expand_func() [6/15] . . . 683
8.64.2.81 expand_func() [7/15] . . . 683
8.64.2.82 expand_func() [8/15] . . . 683
8.64.2.83 expand_func() [9/15] . . . 684
8.64.2.84 expand_func() [10/15] . . . 684
8.64.2.85 expand_func() [11/15] . . . 684
8.64.2.86 expand_func() [12/15] . . . 684
8.64.2.87 expand_func() [13/15] . . . 684
8.64.2.88 expand_func() [14/15] . . . 684
8.64.2.89 derivative_func() [1/15] . . . 684
8.64.2.90 derivative_func() [2/15] . . . 685
8.64.2.91 derivative_func() [3/15] . . . 685
8.64.2.92 derivative_func() [4/15] . . . 685

8.64.2.93 derivative_func() [5/15] . . . 685
8.64.2.94 derivative_func() [6/15] . . . 685
8.64.2.95 derivative_func() [7/15] . . . 685
8.64.2.96 derivative_func() [8/15] . . . 685
8.64.2.97 derivative_func() [9/15] . . . 686
8.64.2.98 derivative_func() [10/15] . . . 686
8.64.2.99 derivative_func() [11/15] . . . 686
8.64.2.100 derivative_func() [12/15] . . . 686
8.64.2.101 derivative_func() [13/15] . . . 686
8.64.2.102 derivative_func() [14/15] . . . 686
8.64.2.103 expl_derivative_func() [1/15] . . . 686
8.64.2.104 expl_derivative_func() [2/15] . . . 687
8.64.2.105 expl_derivative_func() [3/15] . . . 687
8.64.2.106 expl_derivative_func() [4/15] . . . 687
8.64.2.107 expl_derivative_func() [5/15] . . . 687
8.64.2.108 expl_derivative_func() [6/15] . . . 687
8.64.2.109 expl_derivative_func() [7/15] . . . 687
8.64.2.110 expl_derivative_func() [8/15] . . . 687
8.64.2.111 expl_derivative_func() [9/15] . . . 688
8.64.2.112 expl_derivative_func() [10/15] . . . 688
8.64.2.113 expl_derivative_func() [11/15] . . . 688
8.64.2.114 expl_derivative_func() [12/15] . . . 688
8.64.2.115 expl_derivative_func() [13/15] . . . 688
8.64.2.116 expl_derivative_func() [14/15] . . . 688
8.64.2.117 power_func() [1/15] . . . 688
8.64.2.118 power_func() [2/15] . . . 689
8.64.2.119 power_func() [3/15] . . . 689
8.64.2.120 power_func() [4/15] . . . 689
8.64.2.121 power_func() [5/15] . . . 689
8.64.2.122 power_func() [6/15] . . . 689
8.64.2.123 power_func() [7/15] . . . 689
8.64.2.124 power_func() [8/15] . . . 689
8.64.2.125 power_func() [9/15] . . . 690
8.64.2.126 power_func() [10/15] . . . 690
8.64.2.127 power_func() [11/15] . . . 690
8.64.2.128 power_func() [12/15] . . . 690
8.64.2.129 power_func() [13/15] . . . 690
8.64.2.130 power_func() [14/15] . . . 690
8.64.2.131 series_func() [1/15] . . . 690
8.64.2.132 series_func() [2/15] . . . 691
8.64.2.133 series_func() [3/15] . . . 691
8.64.2.134 series_func() [4/15] . . . 691

8.64.2.135 series_func() [5/15] . . . 691
8.64.2.136 series_func() [6/15] . . . 691
8.64.2.137 series_func() [7/15] . . . 691
8.64.2.138 series_func() [8/15] . . . 691
8.64.2.139 series_func() [9/15] . . . 692
8.64.2.140 series_func() [10/15] . . . 692
8.64.2.141 series_func() [11/15] . . . 692
8.64.2.142 series_func() [12/15] . . . 692
8.64.2.143 series_func() [13/15] . . . 692
8.64.2.144 series_func() [14/15] . . . 692
8.64.2.145 info_func() [1/15] . . . 692
8.64.2.146 info_func() [2/15] . . . 693
8.64.2.147 info_func() [3/15] . . . 693
8.64.2.148 info_func() [4/15] . . . 693
8.64.2.149 info_func() [5/15] . . . 693
8.64.2.150 info_func() [6/15] . . . 693
8.64.2.151 info_func() [7/15] . . . 693
8.64.2.152 info_func() [8/15] . . . 693
8.64.2.153 info_func() [9/15] . . . 694
8.64.2.154 info_func() [10/15] . . . 694
8.64.2.155 info_func() [11/15] . . . 694
8.64.2.156 info_func() [12/15] . . . 694
8.64.2.157 info_func() [13/15] . . . 694
8.64.2.158 info_func() [14/15] . . . 694
8.64.2.159 eval_func() [15/15] . . . 694
8.64.2.160 evalf_func() [15/15] . . . 695
8.64.2.161 conjugate_func() [15/15] . . . 695
8.64.2.162 real_part_func() [15/15] . . . 695
8.64.2.163 imag_part_func() [15/15] . . . 695
8.64.2.164 expand_func() [15/15] . . . 695
8.64.2.165 derivative_func() [15/15] . . . 695
8.64.2.166 expl_derivative_func() [15/15] . . . 695
8.64.2.167 power_func() [15/15] . . . 696
8.64.2.168 series_func() [15/15] . . . 696
8.64.2.169 info_func() [15/15] . . . 696
8.64.2.170 print_func() [1/15] . . . 696
8.64.2.171 print_func() [2/15] . . . 696
8.64.2.172 print_func() [3/15] . . . 696
8.64.2.173 print_func() [4/15] . . . 697
8.64.2.174 print_func() [5/15] . . . 697
8.64.2.175 print_func() [6/15] . . . 697
8.64.2.176 print_func() [7/15] . . . 697

8.64.2.177 print_func() [8/15] . . . 697
8.64.2.178 print_func() [9/15] . . . 697
8.64.2.179 print_func() [10/15] . . . 698
8.64.2.180 print_func() [11/15] . . . 698
8.64.2.181 print_func() [12/15] . . . 698
8.64.2.182 print_func() [13/15] . . . 698
8.64.2.183 print_func() [14/15] . . . 698
8.64.2.184 print_func() [15/15] . . . 698
8.64.2.185 set_return_type() . . . 699
8.64.2.186 do_not_evalf_params() . . . 699
8.64.2.187 remember() . . . 699
8.64.2.188 overloaded() . . . 699
8.64.2.189 set_symmetry() . . . 699
8.64.2.190 get_name() . . . 699
8.64.2.191 get_nparams() . . . 699
8.64.2.192 has_derivative() . . . 700
8.64.2.193 has_power() . . . 700
8.64.2.194 test_and_set_nparams() . . . 700
8.64.2.195 set_print_func() . . . 700
8.64.3 Friends And Related Symbol Documentation . . . 700
8.64.3.1 function . . . 700
8.64.3.2 fderivative . . . 701
8.64.4 Member Data Documentation . . . 701
8.64.4.1 name . . . 701
8.64.4.2 TeX_name . . . 701
8.64.4.3 nparams . . . 701
8.64.4.4 eval_f . . . 701
8.64.4.5 evalf_f . . . 701
8.64.4.6 conjugate_f . . . 701
8.64.4.7 real_part_f . . . 702
8.64.4.8 imag_part_f . . . 702
8.64.4.9 expand_f . . . 702
8.64.4.10 derivative_f . . . 702
8.64.4.11 expl_derivative_f . . . 702
8.64.4.12 power_f . . . 702
8.64.4.13 series_f . . . 702
8.64.4.14 print_dispatch_table . . . 702
8.64.4.15 info_f . . . 703
8.64.4.16 evalf_params_first . . . 703
8.64.4.17 use_return_type . . . 703
8.64.4.18 return_type . . . 703
8.64.4.19 return_type_tinfo . . . 703

8.64.4.20 use_remember . . . 703
8.64.4.21 remember_size . . . 703
8.64.4.22 remember_assoc_size . . . 703
8.64.4.23 remember_strategy . . . 704
8.64.4.24 eval_use_exvector_args . . . 704
8.64.4.25 evalf_use_exvector_args . . . 704
8.64.4.26 conjugate_use_exvector_args . . . 704
8.64.4.27 real_part_use_exvector_args . . . 704
8.64.4.28 imag_part_use_exvector_args . . . 704
8.64.4.29 expand_use_exvector_args . . . 704
8.64.4.30 derivative_use_exvector_args . . . 704
8.64.4.31 expl_derivative_use_exvector_args . . . 705
8.64.4.32 power_use_exvector_args . . . 705
8.64.4.33 series_use_exvector_args . . . 705
8.64.4.34 print_use_exvector_args . . . 705
8.64.4.35 info_use_exvector_args . . . 705
8.64.4.36 functions_with_same_name . . . 705
8.64.4.37 symtree . . . 705
8.65 GiNaC::G2_SERIAL Class Reference . . . 706
8.65.1 Detailed Description . . . 706
8.65.2 Member Data Documentation . . . 706
8.65.2.1 serial . . . 706
8.66 GiNaC::G3_SERIAL Class Reference . . . 706
8.66.1 Detailed Description . . . 706
8.66.2 Member Data Documentation . . . 707
8.66.2.1 serial . . . 707
8.67 GiNaC::gcd_options Struct Reference . . . 707
8.67.1 Detailed Description . . . 707
8.67.2 Member Enumeration Documentation . . . 707
8.67.2.1 anonymous enum . . . 707
8.68 GiNaC::gcdheu_failed Class Reference . . . 708
8.68.1 Detailed Description . . . 708
8.69 GiNaC::Gt Class Reference . . . 708
8.69.1 Detailed Description . . . 713
8.69.2 Constructor & Destructor Documentation . . . 713
8.69.2.1 Gt() [1/2] . . . 713
8.69.2.2 Gt() [2/2] . . . 714
8.69.3 Member Function Documentation . . . 714
8.69.3.1 setArgs() . . . 714
8.69.3.2 getTau() . . . 714
8.69.3.3 getZ() . . . 714
8.69.3.4 getArgs() . . . 714

8.69.3.5 precedence() . . . 714
8.69.3.6 eval() . . . 715
8.69.3.7 evalf() . . . 715
8.69.3.8 nops() . . . 715
8.69.3.9 subs() . . . 715
8.69.3.10 calchash() . . . 716
8.69.3.11 has() . . . 716
8.69.3.12 match() . . . 716
8.69.3.13 archive() . . . 716
8.69.3.14 read_archive() . . . 717
8.69.3.15 zisToFundamental() . . . 717
8.69.3.16 regularise() . . . 717
8.69.3.17 findMoebiusTransform() . . . 717
8.69.3.18 tauToFundamental() . . . 717
8.69.3.19 decomposeIntegrationPath() . . . 718
8.69.3.20 applyIntegrationPath() . . . 718
8.69.3.21 qExpand() . . . 718
8.69.3.22 evaluate() . . . 718
8.69.3.23 ex_zisToFundamental() [1/2] . . . 718
8.69.3.24 ex_regularise() [1/2] . . . 719
8.69.3.25 ex_tauToFundamental() [1/2] . . . 719
8.69.3.26 ex_cutIntegrationPath() [1/2] . . . 719
8.69.3.27 ex_qExpand() [1/2] . . . 719
8.69.3.28 ex_prepare() [1/2] . . . 719
8.69.3.29 ex_evaluate() [1/2] . . . 720
8.69.3.30 lst_evaluate() [1/2] . . . 720
8.69.3.31 ex_zisToFundamental() [2/2] . . . 720
8.69.3.32 ex_regularise() [2/2] . . . 720
8.69.3.33 ex_tauToFundamental() [2/2] . . . 720
8.69.3.34 ex_cutIntegrationPath() [2/2] . . . 721
8.69.3.35 ex_qExpand() [2/2] . . . 721
8.69.3.36 ex_prepare() [2/2] . . . 721
8.69.3.37 ex_evaluate() [2/2] . . . 721
8.69.3.38 lst_evaluate() [2/2] . . . 721
8.69.3.39 apply_function_recursive() . . . 721
8.69.3.40 clear_polylog_cache() . . . 722
8.69.3.41 do_print() . . . 722
8.69.3.42 do_print_latex() . . . 722
8.69.3.43 to_numeric() . . . 722
8.69.3.44 to_integer() . . . 722
8.69.4 Member Data Documentation . . . 723
8.69.4.1 cutIntegrationPath_threshold_real . . . 723

8.69.4.2 cutIntegrationPath_threshold_imag . . . 723
8.69.4.3 qExpand_minOrder . . . 723
8.69.4.4 qExpand_stepSize . . . 723
8.69.4.5 auto_clear_polylog_cache . . . 723
8.69.4.6 enable_tauToFundamental . . . 723
8.69.4.7 nargs . . . 723
8.69.4.8 args . . . 724
8.69.4.9 z . . . 724
8.69.4.10 tau . . . 724
8.69.4.11 nqexand_transformer . . . 724
8.70 GiNaC::has_distance< T > Class Template Reference . . . 724
8.70.1 Detailed Description . . . 725
8.70.2 Member Typedef Documentation . . . 725
8.70.2.1 yes_type . . . 725
8.70.2.2 no_type . . . 725
8.70.3 Member Enumeration Documentation . . . 726
8.70.3.1 anonymous enum . . . 726
8.70.4 Member Function Documentation . . . 726
8.70.4.1 test() [1/2] . . . 726
8.70.4.2 test() [2/2] . . . 726
8.71 GiNaC::has_options Class Reference . . . 726
8.71.1 Detailed Description . . . 726
8.71.2 Member Enumeration Documentation . . . 727
8.71.2.1 anonymous enum . . . 727
8.72 std::hash< GiNaC::ex > Struct Reference . . . 727
8.72.1 Detailed Description . . . 727
8.72.2 Member Function Documentation . . . 727
8.72.2.1 operator()() . . . 727
8.73 GiNaC::idx Class Reference . . . 728
8.73.1 Detailed Description . . . 733
8.73.2 Constructor & Destructor Documentation . . . 733
8.73.2.1 idx() . . . 733
8.73.3 Member Function Documentation . . . 734
8.73.3.1 info() . . . 734
8.73.3.2 nops() . . . 734
8.73.3.3 op() . . . 734
8.73.3.4 map() . . . 734
8.73.3.5 evalf() . . . 735
8.73.3.6 subs() . . . 735
8.73.3.7 archive() . . . 735
8.73.3.8 read_archive() . . . 736
8.73.3.9 derivative() . . . 736

8.73.3.10 match_same_type() . . . 736
8.73.3.11 calchash() . . . 737
8.73.3.12 is_dummy_pair_same_type() . . . 737
8.73.3.13 get_value() . . . 737
8.73.3.14 is_numeric() . . . 737
8.73.3.15 is_symbolic() . . . 738
8.73.3.16 get_dim() . . . 738
8.73.3.17 is_dim_numeric() . . . 738
8.73.3.18 is_dim_symbolic() . . . 738
8.73.3.19 replace_dim() . . . 738
8.73.3.20 minimal_dim() . . . 739
8.73.3.21 print_index() . . . 739
8.73.3.22 do_print() . . . 739
8.73.3.23 do_print_csrc() . . . 739
8.73.3.24 do_print_latex() . . . 739
8.73.3.25 do_print_tree() . . . 740
8.73.4 Member Data Documentation . . . 740
8.73.4.1 value . . . 740
8.73.4.2 dim . . . 740
8.74 GiNaC::idx_is_equal_ignore_dim Struct Reference . . . 740
8.74.1 Member Function Documentation . . . 741
8.74.1.1 operator()() . . . 741
8.75 GiNaC::indexed Class Reference . . . 741
8.75.1 Detailed Description . . . 747
8.75.2 Constructor & Destructor Documentation . . . 748
8.75.2.1 indexed() [1/13] . . . 748
8.75.2.2 indexed() [2/13] . . . 748
8.75.2.3 indexed() [3/13] . . . 748
8.75.2.4 indexed() [4/13] . . . 749
8.75.2.5 indexed() [5/13] . . . 749
8.75.2.6 indexed() [6/13] . . . 750
8.75.2.7 indexed() [7/13] . . . 750
8.75.2.8 indexed() [8/13] . . . 751
8.75.2.9 indexed() [9/13] . . . 751
8.75.2.10 indexed() [10/13] . . . 752
8.75.2.11 indexed() [11/13] . . . 752
8.75.2.12 indexed() [12/13] . . . 752
8.75.2.13 indexed() [13/13] . . . 752
8.75.3 Member Function Documentation . . . 752
8.75.3.1 precedence() . . . 752
8.75.3.2 info() . . . 753
8.75.3.3 eval() . . . 753

8.75.3.4 real_part() . . . 753
8.75.3.5 imag_part() . . . 753
8.75.3.6 get_free_indices() . . . 754
8.75.3.7 archive() . . . 754
8.75.3.8 read_archive() . . . 754
8.75.3.9 derivative() . . . 754
8.75.3.10 thiscontainer() [1/2] . . . 755
8.75.3.11 thiscontainer() [2/2] . . . 755
8.75.3.12 return_type() . . . 755
8.75.3.13 return_type_tinfo() . . . 755
8.75.3.14 expand() . . . 756
8.75.3.15 all_index_values_are() . . . 756
8.75.3.16 get_indices() . . . 756
8.75.3.17 get_dummy_indices() [1/2] . . . 756
8.75.3.18 get_dummy_indices() [2/2] . . . 756
8.75.3.19 has_dummy_index_for() . . . 757
8.75.3.20 get_symmetry() . . . 757
8.75.3.21 printindices() . . . 757
8.75.3.22 print_indexed() . . . 757
8.75.3.23 do_print() . . . 757
8.75.3.24 do_print_latex() . . . 758
8.75.3.25 do_print_tree() . . . 758
8.75.3.26 validate() . . . 758
8.75.4 Friends And Related Symbol Documentation . . . 758
8.75.4.1 simplify_indexed . . . 758
8.75.4.2 simplify_indexed_product . . . 759
8.75.4.3 reposition_dummy_indices . . . 759
8.75.5 Member Data Documentation . . . 759
8.75.5.1 symtree . . . 759
8.76 GiNaC::info_flags Class Reference . . . 760
8.76.1 Detailed Description . . . 760
8.76.2 Member Enumeration Documentation . . . 760
8.76.2.1 anonymous enum . . . 760
8.77 GiNaC::integral Class Reference . . . 761
8.77.1 Detailed Description . . . 767
8.77.2 Constructor & Destructor Documentation . . . 767
8.77.2.1 integral() . . . 767
8.77.3 Member Function Documentation . . . 767
8.77.3.1 precedence() . . . 767
8.77.3.2 eval() . . . 767
8.77.3.3 evalf() . . . 768
8.77.3.4 degree() . . . 768

8.77.3.5 ldegree() . . . 768
8.77.3.6 eval_ncmul() . . . 768
8.77.3.7 nops() . . . 768
8.77.3.8 op() . . . 769
8.77.3.9 let_op() . . . 769
8.77.3.10 expand() . . . 769
8.77.3.11 get_free_indices() . . . 769
8.77.3.12 return_type() . . . 770
8.77.3.13 return_type_tinfo() . . . 770
8.77.3.14 conjugate() . . . 770
8.77.3.15 eval_integ() . . . 770
8.77.3.16 archive() . . . 770
8.77.3.17 read_archive() . . . 771
8.77.3.18 derivative() . . . 771
8.77.3.19 series() . . . 771
8.77.3.20 do_print() . . . 772
8.77.3.21 do_print_latex() . . . 772
8.77.4 Member Data Documentation . . . 772
8.77.4.1 max_integration_level . . . 772
8.77.4.2 relative_integration_error . . . 772
8.77.4.3 x . . . 772
8.77.4.4 a . . . 772
8.77.4.5 b . . . 773
8.77.4.6 f . . . 773
8.78 GiNaC::integration_kernel Class Reference . . . 773
8.78.1 Detailed Description . . . 778
8.78.2 Member Function Documentation . . . 778
8.78.2.1 series() . . . 778
8.78.2.2 has_trailing_zero() . . . 778
8.78.2.3 is_numeric() . . . 779
8.78.2.4 Laurent_series() . . . 779
8.78.2.5 get_numerical_value() . . . 779
8.78.2.6 uses_Laurent_series() . . . 780
8.78.2.7 series_coeff_impl() . . . 780
8.78.2.8 get_cache_size() . . . 780
8.78.2.9 set_cache_step() . . . 780
8.78.2.10 get_series_coeff() . . . 781
8.78.2.11 series_coeff() . . . 781
8.78.2.12 get_numerical_value_impl() . . . 781
8.78.2.13 do_print() . . . 781
8.78.3 Member Data Documentation . . . 782
8.78.3.1 cache_step_size . . . 782

8.78.3.2 series_vec . . . 782
8.79 GiNaC::is_not_a_clifford Struct Reference . . . 782
8.79.1 Detailed Description . . . 782
8.79.2 Member Function Documentation . . . 782
8.79.2.1 operator()() . . . 782
8.80 GiNaC::is_summation_idx Struct Reference . . . 783
8.80.1 Member Function Documentation . . . 783
8.80.1.1 operator()() . . . 783
8.81 GiNaC::iterated_integral2_SERIAL Class Reference . . . 783
8.81.1 Detailed Description . . . 783
8.81.2 Member Data Documentation . . . 783
8.81.2.1 serial . . . 783
8.82 GiNaC::iterated_integral3_SERIAL Class Reference . . . 784
8.82.1 Detailed Description . . . 784
8.82.2 Member Data Documentation . . . 784
8.82.2.1 serial . . . 784
8.83 GiNaC::Gt::kernel Struct Reference . . . 784
8.83.1 Constructor & Destructor Documentation . . . 786
8.83.1.1 kernel() [1/3] . . . 786
8.83.1.2 kernel() [2/3] . . . 786
8.83.1.3 kernel() [3/3] . . . 786
8.83.2 Member Function Documentation . . . 786
8.83.2.1 check() . . . 786
8.83.3 Friends And Related Symbol Documentation . . . 786
8.83.3.1 Gt . . . 786
8.83.4 Member Data Documentation . . . 787
8.83.4.1 ni . . . 787
8.83.4.2 zi . . . 787
8.83.4.3 deform . . . 787
8.84 GiNaC::Kronecker_dtau_kernel Class Reference . . . 787
8.84.1 Detailed Description . . . 792
8.84.2 Constructor & Destructor Documentation . . . 793
8.84.2.1 Kronecker_dtau_kernel() . . . 793
8.84.3 Member Function Documentation . . . 793
8.84.3.1 nops() . . . 793
8.84.3.2 op() . . . 793
8.84.3.3 let_op() . . . 793
8.84.3.4 is_numeric() . . . 794
8.84.3.5 get_numerical_value() . . . 794
8.84.3.6 series_coeff_impl() . . . 794
8.84.3.7 do_print() . . . 794
8.84.4 Member Data Documentation . . . 795

8.84.4.1 n . . . 795
8.84.4.2 z . . . 795
8.84.4.3 K . . . 795
8.84.4.4 C_norm . . . 795
8.85 GiNaC::Kronecker_dz_kernel Class Reference . . . 796
8.85.1 Detailed Description . . . 801
8.85.2 Constructor & Destructor Documentation . . . 802
8.85.2.1 Kronecker_dz_kernel() . . . 802
8.85.3 Member Function Documentation . . . 802
8.85.3.1 nops() . . . 802
8.85.3.2 op() . . . 802
8.85.3.3 let_op() . . . 802
8.85.3.4 is_numeric() . . . 803
8.85.3.5 get_numerical_value() . . . 803
8.85.3.6 series_coeff_impl() . . . 803
8.85.3.7 do_print() . . . 803
8.85.4 Member Data Documentation . . . 804
8.85.4.1 n . . . 804
8.85.4.2 z_j . . . 804
8.85.4.3 tau . . . 804
8.85.4.4 K . . . 804
8.85.4.5 C_norm . . . 804
8.86 GiNaC::lanczos_coeffs Class Reference . . . 804
8.86.1 Constructor & Destructor Documentation . . . 805
8.86.1.1 lanczos_coeffs() . . . 805
8.86.2 Member Function Documentation . . . 805
8.86.2.1 sufficiently_accurate() . . . 805
8.86.2.2 get_order() . . . 805
8.86.2.3 calc_lanczos_A() . . . 805
8.86.3 Member Data Documentation . . . 806
8.86.3.1 coeffs . . . 806
8.86.3.2 current_vector . . . 806
8.87 std::less< GiNaC::ptr< T > > Struct Template Reference . . . 806
8.87.1 Detailed Description . . . 806
8.87.2 Member Function Documentation . . . 806
8.87.2.1 operator()() . . . 806
8.88 GiNaC::library_init Class Reference . . . 807
8.88.1 Detailed Description . . . 807
8.88.2 Constructor & Destructor Documentation . . . 808
8.88.2.1 library_init() . . . 808
8.88.2.2 ~library_init() . . . 808
8.88.3 Member Function Documentation . . . 808

8.88.3.1 init_unarchivers() . . . 808
8.88.4 Member Data Documentation . . . 809
8.88.4.1 count . . . 809
8.89 GiNaC::make_flat_inserter Class Reference . . . 809
8.89.1 Detailed Description . . . 809
8.89.2 Constructor & Destructor Documentation . . . 810
8.89.2.1 make_flat_inserter() [1/2] . . . 810
8.89.2.2 make_flat_inserter() [2/2] . . . 810
8.89.3 Member Function Documentation . . . 810
8.89.3.1 handle_factor() . . . 810
8.89.3.2 combine_indices() . . . 810
8.89.4 Member Data Documentation . . . 810
8.89.4.1 do_renaming . . . 810
8.89.4.2 used_indices . . . 811
8.90 GiNaC::map_function Struct Reference . . . 811
8.90.1 Detailed Description . . . 813
8.90.2 Member Typedef Documentation . . . 813
8.90.2.1 argument_type . . . 813
8.90.2.2 result_type . . . 813
8.90.3 Constructor & Destructor Documentation . . . 813
8.90.3.1 ∼map_function() . . . 813
8.90.4 Member Function Documentation . . . 813
8.90.4.1 operator()() . . . 813
8.91 GiNaC::matrix Class Reference . . . 814
8.91.1 Detailed Description . . . 819
8.91.2 Constructor & Destructor Documentation . . . 819
8.91.2.1 matrix() [1/5] . . . 819
8.91.2.2 matrix() [2/5] . . . 820
8.91.2.3 matrix() [3/5] . . . 820
8.91.2.4 matrix() [4/5] . . . 820
8.91.2.5 matrix() [5/5] . . . 820
8.91.3 Member Function Documentation . . . 821
8.91.3.1 nops() . . . 821
8.91.3.2 op() . . . 821
8.91.3.3 let_op() . . . 821
8.91.3.4 evalm() . . . 821
8.91.3.5 subs() . . . 822
8.91.3.6 eval_indexed() . . . 822
8.91.3.7 add_indexed() . . . 822
8.91.3.8 scalar_mul_indexed() . . . 822
8.91.3.9 contract_with() . . . 823
8.91.3.10 conjugate() . . . 823

8.91.3.11 real_part() . . . 823
8.91.3.12 imag_part() . . . 823
8.91.3.13 archive() . . . 823
8.91.3.14 read_archive() . . . 824
8.91.3.15 match_same_type() . . . 824
8.91.3.16 return_type() . . . 824
8.91.3.17 rows() . . . 824
8.91.3.18 cols() . . . 825
8.91.3.19 add() . . . 825
8.91.3.20 sub() . . . 825
8.91.3.21 mul() [1/2] . . . 826
8.91.3.22 mul() [2/2] . . . 826
8.91.3.23 mul_scalar() . . . 826
8.91.3.24 pow() . . . 826
8.91.3.25 operator()() [1/2] . . . 827
8.91.3.26 operator()() [2/2] . . . 827
8.91.3.27 set() . . . 827
8.91.3.28 transpose() . . . 828
8.91.3.29 determinant() . . . 828
8.91.3.30 trace() . . . 829
8.91.3.31 charpoly() . . . 829
8.91.3.32 inverse() [1/2] . . . 830
8.91.3.33 inverse() [2/2] . . . 830
8.91.3.34 solve() . . . 830
8.91.3.35 rank() [1/2] . . . 831
8.91.3.36 rank() [2/2] . . . 831
8.91.3.37 is_zero_matrix() . . . 831
8.91.3.38 determinant_minor() . . . 832
8.91.3.39 echelon_form() . . . 832
8.91.3.40 gauss_elimination() . . . 832
8.91.3.41 division_free_elimination() . . . 833
8.91.3.42 fraction_free_elimination() . . . 833
8.91.3.43 markowitz_elimination() . . . 834
8.91.3.44 pivot() . . . 834
8.91.3.45 print_elements() . . . 834
8.91.3.46 do_print() . . . 835
8.91.3.47 do_print_latex() . . . 835
8.91.3.48 do_print_python_repr() . . . 835
8.91.4 Member Data Documentation . . . 835
8.91.4.1 row . . . 835
8.91.4.2 col . . . 835
8.91.4.3 m . . . 836

8.92 GiNaC::minkmetric Class Reference . . . 836
8.92.1 Detailed Description . . . 841
8.92.2 Constructor & Destructor Documentation . . . 841
8.92.2.1 minkmetric() . . . 841
8.92.3 Member Function Documentation . . . 841
8.92.3.1 info() . . . 841
8.92.3.2 eval_indexed() . . . 842
8.92.3.3 archive() . . . 842
8.92.3.4 read_archive() . . . 842
8.92.3.5 return_type() . . . 842
8.92.3.6 do_print() . . . 842
8.92.3.7 do_print_latex() . . . 843
8.92.4 Member Data Documentation . . . 843
8.92.4.1 pos_sig . . . 843
8.93 GiNaC::modular_form_kernel Class Reference . . . 843
8.93.1 Detailed Description . . . 848
8.93.2 Constructor & Destructor Documentation . . . 849
8.93.2.1 modular_form_kernel() . . . 849
8.93.3 Member Function Documentation . . . 849
8.93.3.1 series() . . . 849
8.93.3.2 nops() . . . 849
8.93.3.3 op() . . . 849
8.93.3.4 let_op() . . . 850
8.93.3.5 is_numeric() . . . 850
8.93.3.6 Laurent_series() . . . 850
8.93.3.7 get_numerical_value() . . . 850
8.93.3.8 uses_Laurent_series() . . . 851
8.93.3.9 q_expansion_modular_form() . . . 851
8.93.3.10 do_print() . . . 851
8.93.4 Member Data Documentation . . . 851
8.93.4.1 k . . . 851
8.93.4.2 P . . . 851
8.93.4.3 C_norm . . . 852
8.94 GiNaC::basic_partition_generator::mpartition2 Struct Reference . . . 852
8.94.1 Constructor & Destructor Documentation . . . 852
8.94.1.1 mpartition2() . . . 852
8.94.2 Member Function Documentation . . . 852
8.94.2.1 next_partition() . . . 852
8.94.3 Member Data Documentation . . . 853
8.94.3.1 x . . . 853
8.94.3.2 n . . . 853
8.94.3.3 m . . . 853

8.95 GiNaC::mul Class Reference . . . 853
8.95.1 Detailed Description . . . 859
8.95.2 Constructor & Destructor Documentation . . . 859
8.95.2.1 mul() [1/7] . . . 859
8.95.2.2 mul() [2/7] . . . 859
8.95.2.3 mul() [3/7] . . . 859
8.95.2.4 mul() [4/7] . . . 860
8.95.2.5 mul() [5/7] . . . 860
8.95.2.6 mul() [6/7] . . . 860
8.95.2.7 mul() [7/7] . . . 860
8.95.3 Member Function Documentation . . . 860
8.95.3.1 precedence() . . . 860
8.95.3.2 info() . . . 861
8.95.3.3 is_polynomial() . . . 861
8.95.3.4 degree() . . . 861
8.95.3.5 ldegree() . . . 861
8.95.3.6 coeff() . . . 862
8.95.3.7 has() . . . 862
8.95.3.8 eval() . . . 862
8.95.3.9 evalf() . . . 863
8.95.3.10 real_part() . . . 863
8.95.3.11 imag_part() . . . 863
8.95.3.12 evalm() . . . 863
8.95.3.13 series() . . . 863
8.95.3.14 normal() . . . 864
8.95.3.15 integer_content() . . . 864
8.95.3.16 smod() . . . 864
8.95.3.17 max_coefficient() . . . 865
8.95.3.18 get_free_indices() . . . 865
8.95.3.19 conjugate() . . . 865
8.95.3.20 derivative() . . . 865
8.95.3.21 eval_ncmul() . . . 866
8.95.3.22 return_type() . . . 866
8.95.3.23 return_type_tinfo() . . . 866
8.95.3.24 thisexpairseq() [1/2] . . . 866
8.95.3.25 thisexpairseq() [2/2] . . . 866
8.95.3.26 split_ex_to_pair() . . . 867
8.95.3.27 combine_ex_with_coeff_to_pair() . . . 867
8.95.3.28 combine_pair_with_coeff_to_pair() . . . 867
8.95.3.29 recombine_pair_to_ex() . . . 867
8.95.3.30 expair_needs_further_processing() . . . 868
8.95.3.31 default_overall_coeff() . . . 868

8.95.3.32 combine_overall_coeff() [1/2] . . . 868
8.95.3.33 combine_overall_coeff() [2/2] . . . 868
8.95.3.34 can_make_flat() . . . 868
8.95.3.35 expand() . . . 869
8.95.3.36 algebraic_subs_mul() . . . 869
8.95.3.37 find_real_imag() . . . 869
8.95.3.38 print_overall_coeff() . . . 869
8.95.3.39 do_print() . . . 870
8.95.3.40 do_print_latex() . . . 870
8.95.3.41 do_print_csrc() . . . 870
8.95.3.42 do_print_python_repr() . . . 870
8.95.3.43 can_be_further_expanded() . . . 870
8.95.3.44 expandchildren() . . . 871
8.95.4 Friends And Related Symbol Documentation . . . 871
8.95.4.1 add . . . 871
8.95.4.2 ncmul . . . 871
8.95.4.3 power . . . 871
8.96 GiNaC::multi_iterator_counter< T > Class Template Reference . . . 872
8.96.1 Detailed Description . . . 873
8.96.2 Constructor & Destructor Documentation . . . 874
8.96.2.1 multi_iterator_counter() [1/3] . . . 874
8.96.2.2 multi_iterator_counter() [2/3] . . . 874
8.96.2.3 multi_iterator_counter() [3/3] . . . 874
8.96.3 Member Function Documentation . . . 874
8.96.3.1 init() . . . 874
8.96.3.2 operator++() . . . 875
8.96.4 Friends And Related Symbol Documentation . . . 875
8.96.4.1 operator<< . . . 875
8.97 GiNaC::multi_iterator_counter_indv< T > Class Template Reference . . . 875
8.97.1 Detailed Description . . . 877
8.97.2 Constructor & Destructor Documentation . . . 877
8.97.2.1 multi_iterator_counter_indv() [1/3] . . . 877
8.97.2.2 multi_iterator_counter_indv() [2/3] . . . 878
8.97.2.3 multi_iterator_counter_indv() [3/3] . . . 878
8.97.3 Member Function Documentation . . . 878
8.97.3.1 init() . . . 878
8.97.3.2 operator++() . . . 878
8.97.4 Friends And Related Symbol Documentation . . . 879
8.97.4.1 operator<< . . . 879
8.97.5 Member Data Documentation . . . 879
8.97.5.1 Nv . . . 879
8.98 GiNaC::multi_iterator_ordered< T > Class Template Reference . . . 879

8.98.1 Detailed Description . . . 881
8.98.2 Constructor & Destructor Documentation . . . 881
8.98.2.1 multi_iterator_ordered() [1/3] . . . 881
8.98.2.2 multi_iterator_ordered() [2/3] . . . 882
8.98.2.3 multi_iterator_ordered() [3/3] . . . 882
8.98.3 Member Function Documentation . . . 882
8.98.3.1 init() . . . 882
8.98.3.2 operator++() . . . 882
8.98.4 Friends And Related Symbol Documentation . . . 883
8.98.4.1 operator<< . . . 883
8.99 GiNaC::multi_iterator_ordered_eq< T > Class Template Reference . . . 883
8.99.1 Detailed Description . . . 885
8.99.2 Constructor & Destructor Documentation . . . 885
8.99.2.1 multi_iterator_ordered_eq() [1/3] . . . 885
8.99.2.2 multi_iterator_ordered_eq() [2/3] . . . 885
8.99.2.3 multi_iterator_ordered_eq() [3/3] . . . 885
8.99.3 Member Function Documentation . . . 886
8.99.3.1 init() . . . 886
8.99.3.2 operator++() . . . 886
8.99.4 Friends And Related Symbol Documentation . . . 886
8.99.4.1 operator<< . . . 886
8.100 GiNaC::multi_iterator_ordered_eq_indv< T > Class Template Reference . . . 887
8.100.1 Detailed Description . . . 888
8.100.2 Constructor & Destructor Documentation . . . 889
8.100.2.1 multi_iterator_ordered_eq_indv() [1/3] . . . 889
8.100.2.2 multi_iterator_ordered_eq_indv() [2/3] . . . 889
8.100.2.3 multi_iterator_ordered_eq_indv() [3/3] . . . 889
8.100.3 Member Function Documentation . . . 889
8.100.3.1 init() . . . 889
8.100.3.2 operator++() . . . 890
8.100.4 Friends And Related Symbol Documentation . . . 890
8.100.4.1 operator<< . . . 890
8.100.5 Member Data Documentation . . . 890
8.100.5.1 Nv . . . 890
8.101 GiNaC::multi_iterator_permutation< T > Class Template Reference . . . 891
8.101.1 Detailed Description . . . 893
8.101.2 Constructor & Destructor Documentation . . . 893
8.101.2.1 multi_iterator_permutation() [1/3] . . . 893
8.101.2.2 multi_iterator_permutation() [2/3] . . . 893
8.101.2.3 multi_iterator_permutation() [3/3] . . . 893
8.101.3 Member Function Documentation . . . 894
8.101.3.1 init() . . . 894

8.101.3.2 operator++() . . . 894
8.101.3.3 get_sign() . . . 894
8.101.4 Friends And Related Symbol Documentation . . . 895
8.101.4.1 operator<< . . . 895
8.102 GiNaC::multi_iterator_shuffle< T > Class Template Reference . . . 895
8.102.1 Detailed Description . . . 897
8.102.2 Constructor & Destructor Documentation . . . 897
8.102.2.1 multi_iterator_shuffle() [1/2] . . . 897
8.102.2.2 multi_iterator_shuffle() [2/2] . . . 897
8.102.3 Member Function Documentation . . . 898
8.102.3.1 init() . . . 898
8.102.3.2 operator++() . . . 898
8.102.4 Friends And Related Symbol Documentation . . . 898
8.102.4.1 operator<< . . . 898
8.102.5 Member Data Documentation . . . 898
8.102.5.1 N_internal . . . 898
8.102.5.2 v_internal . . . 899
8.102.5.3 v_orig . . . 899
8.103 GiNaC::multi_iterator_shuffle_prime< T > Class Template Reference . . . 899
8.103.1 Detailed Description . . . 901
8.103.2 Constructor & Destructor Documentation . . . 902
8.103.2.1 multi_iterator_shuffle_prime() [1/2] . . . 902
8.103.2.2 multi_iterator_shuffle_prime() [2/2] . . . 902
8.103.3 Member Function Documentation . . . 902
8.103.3.1 init() . . . 902
8.103.4 Friends And Related Symbol Documentation . . . 902
8.103.4.1 operator<< . . . 902
8.104 GiNaC::multiple_polylog_kernel Class Reference . . . 903
8.104.1 Detailed Description . . . 908
8.104.2 Constructor & Destructor Documentation . . . 909
8.104.2.1 multiple_polylog_kernel() . . . 909
8.104.3 Member Function Documentation . . . 909
8.104.3.1 nops() . . . 909
8.104.3.2 op() . . . 909
8.104.3.3 let_op() . . . 909
8.104.3.4 is_numeric() . . . 909
8.104.3.5 series_coeff_impl() . . . 910
8.104.3.6 do_print() . . . 910
8.104.4 Member Data Documentation . . . 910
8.104.4.1 z . . . 910
8.105 GiNaC::ncmul Class Reference . . . 911
8.105.1 Detailed Description . . . 916

8.105.2 Constructor & Destructor Documentation . . . 917
8.105.2.1 ncmul() [1/7] . . . 917
8.105.2.2 ncmul() [2/7] . . . 917
8.105.2.3 ncmul() [3/7] . . . 917
8.105.2.4 ncmul() [4/7] . . . 917
8.105.2.5 ncmul() [5/7] . . . 917
8.105.2.6 ncmul() [6/7] . . . 917
8.105.2.7 ncmul() [7/7] . . . 918
8.105.3 Member Function Documentation . . . 918
8.105.3.1 precedence() . . . 918
8.105.3.2 info() . . . 918
8.105.3.3 degree() . . . 918
8.105.3.4 ldegree() . . . 918
8.105.3.5 expand() . . . 919
8.105.3.6 coeff() . . . 919
8.105.3.7 eval() . . . 919
8.105.3.8 evalm() . . . 920
8.105.3.9 get_free_indices() . . . 920
8.105.3.10 thiscontainer() [1/2] . . . 920
8.105.3.11 thiscontainer() [2/2] . . . 920
8.105.3.12 conjugate() . . . 921
8.105.3.13 real_part() . . . 921
8.105.3.14 imag_part() . . . 921
8.105.3.15 derivative() . . . 921
8.105.3.16 return_type() . . . 921
8.105.3.17 return_type_tinfo() . . . 922
8.105.3.18 do_print() . . . 922
8.105.3.19 do_print_csrc() . . . 922
8.105.3.20 count_factors() . . . 922
8.105.3.21 append_factors() . . . 922
8.105.3.22 expandchildren() . . . 923
8.105.3.23 get_factors() . . . 923
8.105.4 Friends And Related Symbol Documentation . . . 923
8.105.4.1 power . . . 923
8.105.4.2 reeval_ncmul . . . 923
8.105.4.3 hold_ncmul . . . 923
8.106 GiNaC::normal_map_function Struct Reference . . . 924
8.106.1 Detailed Description . . . 925
8.106.2 Member Function Documentation . . . 925
8.106.2.1 operator()() . . . 925
8.107 GiNaC::numeric Class Reference . . . 925
8.107.1 Detailed Description . . . 931

8.107.2 Constructor & Destructor Documentation . . . 932
8.107.2.1 numeric() [1/10] . . . 932
8.107.2.2 numeric() [2/10] . . . 932
8.107.2.3 numeric() [3/10] . . . 932
8.107.2.4 numeric() [4/10] . . . 932
8.107.2.5 numeric() [5/10] . . . 933
8.107.2.6 numeric() [6/10] . . . 933
8.107.2.7 numeric() [7/10] . . . 933
8.107.2.8 numeric() [8/10] . . . 933
8.107.2.9 numeric() [9/10] . . . 933
8.107.2.10 numeric() [10/10] . . . 934
8.107.3 Member Function Documentation . . . 934
8.107.3.1 precedence() . . . 934
8.107.3.2 info() . . . 934
8.107.3.3 is_polynomial() . . . 935
8.107.3.4 degree() . . . 935
8.107.3.5 ldegree() . . . 935
8.107.3.6 coeff() . . . 935
8.107.3.7 has() . . . 936
8.107.3.8 eval() . . . 936
8.107.3.9 evalf() . . . 936
8.107.3.10 subs() . . . 937
8.107.3.11 normal() . . . 937
8.107.3.12 to_rational() . . . 937
8.107.3.13 to_polynomial() . . . 938
8.107.3.14 integer_content() . . . 938
8.107.3.15 smod() . . . 938
8.107.3.16 max_coefficient() . . . 939
8.107.3.17 conjugate() . . . 939
8.107.3.18 real_part() . . . 939
8.107.3.19 imag_part() . . . 939
8.107.3.20 archive() . . . 939
8.107.3.21 read_archive() . . . 940
8.107.3.22 derivative() . . . 940
8.107.3.23 is_equal_same_type() . . . 940
8.107.3.24 calchash() . . . 940
8.107.3.25 add() . . . 941
8.107.3.26 sub() . . . 941
8.107.3.27 mul() . . . 941
8.107.3.28 div() . . . 942
8.107.3.29 power() . . . 942
8.107.3.30 add_dyn() . . . 942

8.107.3.31 sub_dyn() . . . 942
8.107.3.32 mul_dyn() . . . 943
8.107.3.33 div_dyn() . . . 943
8.107.3.34 power_dyn() . . . 943
8.107.3.35 operator=() [1/6] . . . 943
8.107.3.36 operator=() [2/6] . . . 944
8.107.3.37 operator=() [3/6] . . . 944
8.107.3.38 operator=() [4/6] . . . 944
8.107.3.39 operator=() [5/6] . . . 944
8.107.3.40 operator=() [6/6] . . . 944
8.107.3.41 inverse() . . . 944
8.107.3.42 step() . . . 945
8.107.3.43 csgn() . . . 945
8.107.3.44 compare() . . . 945
8.107.3.45 is_equal() . . . 946
8.107.3.46 is_zero() . . . 946
8.107.3.47 is_positive() . . . 946
8.107.3.48 is_negative() . . . 946
8.107.3.49 is_integer() . . . 947
8.107.3.50 is_pos_integer() . . . 947
8.107.3.51 is_nonneg_integer() . . . 947
8.107.3.52 is_even() . . . 947
8.107.3.53 is_odd() . . . 947
8.107.3.54 is_prime() . . . 948
8.107.3.55 is_rational() . . . 948
8.107.3.56 is_real() . . . 948
8.107.3.57 is_cinteger() . . . 948
8.107.3.58 is_crational() . . . 949
8.107.3.59 operator==() . . . 949
8.107.3.60 operator"!=() . . . 949
8.107.3.61 operator<() . . . 949
8.107.3.62 operator<=() . . . 949
8.107.3.63 operator>() . . . 950
8.107.3.64 operator>=() . . . 950
8.107.3.65 to_int() . . . 950
8.107.3.66 to_long() . . . 950
8.107.3.67 to_double() . . . 951
8.107.3.68 to_cl_N() . . . 951
8.107.3.69 real() . . . 951
8.107.3.70 imag() . . . 951
8.107.3.71 numer() . . . 952
8.107.3.72 denom() . . . 952

8.107.3.73 int_length() . . . 952
8.107.3.74 print_numeric() . . . 952
8.107.3.75 do_print() . . . 953
8.107.3.76 do_print_latex() . . . 953
8.107.3.77 do_print_csrc() . . . 953
8.107.3.78 do_print_csrc_cl_N() . . . 953
8.107.3.79 do_print_tree() . . . 953
8.107.3.80 do_print_python_repr() . . . 954
8.107.4 Member Data Documentation . . . 954
8.107.4.1 value . . . 954
8.108 GiNaC::op0_is_equal Struct Reference . . . 954
8.108.1 Member Function Documentation . . . 954
8.108.1.1 operator()() . . . 954
8.109 GiNaC::partition_generator Class Reference . . . 955
8.109.1 Detailed Description . . . 956
8.109.2 Constructor & Destructor Documentation . . . 956
8.109.2.1 partition_generator() . . . 956
8.109.3 Member Function Documentation . . . 956
8.109.3.1 get() . . . 956
8.109.3.2 next() . . . 956
8.109.4 Member Data Documentation . . . 957
8.109.4.1 partition . . . 957
8.109.4.2 current_updated . . . 957
8.110 GiNaC::partition_with_zero_parts_generator Class Reference . . . 957
8.110.1 Detailed Description . . . 958
8.110.2 Constructor & Destructor Documentation . . . 959
8.110.2.1 partition_with_zero_parts_generator() . . . 959
8.110.3 Member Function Documentation . . . 959
8.110.3.1 get() . . . 959
8.110.3.2 next() . . . 959
8.110.4 Member Data Documentation . . . 959
8.110.4.1 m . . . 959
8.110.4.2 partition . . . 959
8.110.4.3 current_updated . . . 960
8.111 GiNaC::Gt_detail::pathintegral_term Class Reference . . . 960
8.111.1 Constructor & Destructor Documentation . . . 961
8.111.1.1 pathintegral_term() [1/2] . . . 961
8.111.1.2 pathintegral_term() [2/2] . . . 961
8.111.2 Member Function Documentation . . . 962
8.111.2.1 integrate() . . . 962
8.111.2.2 evaluate() . . . 962
8.111.2.3 G_path() [1/2] . . . 962

8.111.2.4 G_path() [2/2] . . . 962
8.111.3 Friends And Related Symbol Documentation . . . 962
8.111.3.1 pathintegral_term_list . . . 962
8.111.3.2 operator<< . . . 963
8.111.3.3 operator< . . . 963
8.111.4 Member Data Documentation . . . 963
8.111.4.1 prefactor . . . 963
8.111.4.2 power . . . 963
8.111.4.3 polylog . . . 963
8.111.4.4 denom . . . 963
8.112 GiNaC::Gt_detail::pathintegral_term_list Class Reference . . . 964
8.112.1 Member Function Documentation . . . 964
8.112.1.1 begin() . . . 964
8.112.1.2 end() . . . 964
8.112.1.3 clear() . . . 964
8.112.1.4 empty() . . . 965
8.112.1.5 size() . . . 965
8.112.1.6 add() [1/3] . . . 965
8.112.1.7 add() [2/3] . . . 965
8.112.1.8 add() [3/3] . . . 965
8.112.1.9 pop() . . . 965
8.112.2 Member Data Documentation . . . 966
8.112.2.1 terms . . . 966
8.113 GiNaC::pointer_to_map_function Class Reference . . . 966
8.113.1 Constructor & Destructor Documentation . . . 968
8.113.1.1 pointer_to_map_function() . . . 968
8.113.2 Member Function Documentation . . . 968
8.113.2.1 operator()() . . . 968
8.113.3 Member Data Documentation . . . 968
8.113.3.1 ptr . . . 968
8.114 GiNaC::pointer_to_map_function_1arg< T1 > Class Template Reference . . . 968
8.114.1 Constructor & Destructor Documentation . . . 970
8.114.1.1 pointer_to_map_function_1arg() . . . 970
8.114.2 Member Function Documentation . . . 970
8.114.2.1 operator()() . . . 970
8.114.3 Member Data Documentation . . . 970
8.114.3.1 ptr . . . 970
8.114.3.2 arg1 . . . 970
8.115 GiNaC::pointer_to_map_function_2args< T1, T2 > Class Template Reference . . . 971
8.115.1 Constructor & Destructor Documentation . . . 972
8.115.1.1 pointer_to_map_function_2args() . . . 972
8.115.2 Member Function Documentation . . . 972

8.115.2.1 operator()() . . . 972
8.115.3 Member Data Documentation . . . 973
8.115.3.1 ptr . . . 973
8.115.3.2 arg1 . . . 973
8.115.3.3 arg2 . . . 973
8.116 GiNaC::pointer_to_map_function_3args< T1, T2, T3 > Class Template Reference . . . 973
8.116.1 Constructor & Destructor Documentation . . . 975
8.116.1.1 pointer_to_map_function_3args() . . . 975
8.116.2 Member Function Documentation . . . 975
8.116.2.1 operator()() . . . 975
8.116.3 Member Data Documentation . . . 975
8.116.3.1 ptr . . . 975
8.116.3.2 arg1 . . . 975
8.116.3.3 arg2 . . . 976
8.116.3.4 arg3 . . . 976
8.117 GiNaC::pointer_to_member_to_map_function< C > Class Template Reference . . . 976
8.117.1 Constructor & Destructor Documentation . . . 978
8.117.1.1 pointer_to_member_to_map_function() . . . 978
8.117.2 Member Function Documentation . . . 978
8.117.2.1 operator()() . . . 978
8.117.3 Member Data Documentation . . . 978
8.117.3.1 ptr . . . 978
8.117.3.2 c . . . 978
8.118 GiNaC::pointer_to_member_to_map_function_1arg< C, T1 > Class Template Reference . . . 979
8.118.1 Constructor & Destructor Documentation . . . 981
8.118.1.1 pointer_to_member_to_map_function_1arg() . . . 981
8.118.2 Member Function Documentation . . . 981
8.118.2.1 operator()() . . . 981
8.118.3 Member Data Documentation . . . 981
8.118.3.1 ptr . . . 981
8.118.3.2 c . . . 981
8.118.3.3 arg1 . . . 982
8.119 GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 > Class Template Reference . . . 982
8.119.1 Constructor & Destructor Documentation . . . 984
8.119.1.1 pointer_to_member_to_map_function_2args() . . . 984
8.119.2 Member Function Documentation . . . 984
8.119.2.1 operator()() . . . 984
8.119.3 Member Data Documentation . . . 984
8.119.3.1 ptr . . . 984
8.119.3.2 c . . . 984
8.119.3.3 arg1 . . . 985
8.119.3.4 arg2 . . . 985

8.120 GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 > Class Template Reference 985
8.120.1 Constructor & Destructor Documentation . . . 987
8.120.1.1 pointer_to_member_to_map_function_3args() . . . 987
8.120.2 Member Function Documentation . . . 987
8.120.2.1 operator()() . . . 987
8.120.3 Member Data Documentation . . . 987
8.120.3.1 ptr . . . 987
8.120.3.2 c . . . 987
8.120.3.3 arg1 . . . 988
8.120.3.4 arg2 . . . 988
8.120.3.5 arg3 . . . 988
8.121 GiNaC::pole_error Class Reference . . . 988
8.121.1 Detailed Description . . . 989
8.121.2 Constructor & Destructor Documentation . . . 989
8.121.2.1 pole_error() . . . 989
8.121.3 Member Function Documentation . . . 990
8.121.3.1 degree() . . . 990
8.121.4 Member Data Documentation . . . 990
8.121.4.1 deg . . . 990
8.122 GiNaC::possymbol Class Reference . . . 990
8.122.1 Detailed Description . . . 995
8.122.2 Constructor & Destructor Documentation . . . 995
8.122.2.1 possymbol() [1/3] . . . 995
8.122.2.2 possymbol() [2/3] . . . 995
8.122.2.3 possymbol() [3/3] . . . 995
8.122.3 Member Function Documentation . . . 996
8.122.3.1 get_domain() . . . 996
8.122.3.2 duplicate() . . . 996
8.123 GiNaC::power Class Reference . . . 996
8.123.1 Detailed Description . . . 1001
8.123.2 Constructor & Destructor Documentation . . . 1001
8.123.2.1 power() [1/2] . . . 1001
8.123.2.2 power() [2/2] . . . 1001
8.123.3 Member Function Documentation . . . 1001
8.123.3.1 precedence() . . . 1001
8.123.3.2 info() . . . 1002
8.123.3.3 nops() . . . 1002
8.123.3.4 op() . . . 1002
8.123.3.5 map() . . . 1002
8.123.3.6 is_polynomial() . . . 1003
8.123.3.7 degree() . . . 1003
8.123.3.8 ldegree() . . . 1003

8.123.3.9 coeff() . . . 1003
8.123.3.10 eval() . . . 1004
8.123.3.11 evalf() . . . 1004
8.123.3.12 evalm() . . . 1005
8.123.3.13 series() . . . 1005
8.123.3.14 subs() . . . 1005
8.123.3.15 has() . . . 1006
8.123.3.16 normal() . . . 1006
8.123.3.17 to_rational() . . . 1006
8.123.3.18 to_polynomial() . . . 1007
8.123.3.19 conjugate() . . . 1007
8.123.3.20 real_part() . . . 1007
8.123.3.21 imag_part() . . . 1007
8.123.3.22 archive() . . . 1007
8.123.3.23 read_archive() . . . 1008
8.123.3.24 derivative() . . . 1008
8.123.3.25 eval_ncmul() . . . 1008
8.123.3.26 return_type() . . . 1008
8.123.3.27 return_type_tinfo() . . . 1009
8.123.3.28 expand() . . . 1009
8.123.3.29 print_power() . . . 1009
8.123.3.30 do_print_dflt() . . . 1009
8.123.3.31 do_print_latex() . . . 1010
8.123.3.32 do_print_csrc() . . . 1010
8.123.3.33 do_print_python() . . . 1010
8.123.3.34 do_print_python_repr() . . . 1010
8.123.3.35 do_print_csrc_cl_N() . . . 1010
8.123.3.36 expand_add() . . . 1011
8.123.3.37 expand_add_2() . . . 1011
8.123.3.38 expand_mul() . . . 1012
8.123.4 Friends And Related Symbol Documentation . . . 1012
8.123.4.1 mul . . . 1012
8.123.5 Member Data Documentation . . . 1012
8.123.5.1 basis . . . 1012
8.123.5.2 exponent . . . 1013
8.124 GiNaC::print_context Class Reference . . . 1013
8.124.1 Detailed Description . . . 1014
8.124.2 Constructor & Destructor Documentation . . . 1014
8.124.2.1 print_context() . . . 1014
8.124.2.2 ∼print_context() . . . 1014
8.124.3 Member Data Documentation . . . 1014
8.124.3.1 s . . . 1014

8.124.3.2 options . . . 1015
8.125 GiNaC::print_context_options Class Reference . . . 1015
8.125.1 Detailed Description . . . 1016
8.125.2 Constructor & Destructor Documentation . . . 1016
8.125.2.1 print_context_options() . . . 1016
8.125.3 Member Function Documentation . . . 1016
8.125.3.1 get_name() . . . 1016
8.125.3.2 get_parent_name() . . . 1016
8.125.3.3 get_id() . . . 1016
8.125.4 Member Data Documentation . . . 1016
8.125.4.1 name . . . 1016
8.125.4.2 parent_name . . . 1017
8.125.4.3 id . . . 1017
8.126 GiNaC::print_csrc Class Reference . . . 1017
8.126.1 Detailed Description . . . 1018
8.126.2 Constructor & Destructor Documentation . . . 1018
8.126.2.1 print_csrc() . . . 1018
8.127 GiNaC::print_csrc_cl_N Class Reference . . . 1019
8.127.1 Detailed Description . . . 1020
8.127.2 Constructor & Destructor Documentation . . . 1020
8.127.2.1 print_csrc_cl_N() . . . 1020
8.128 GiNaC::print_csrc_double Class Reference . . . 1021
8.128.1 Detailed Description . . . 1022
8.128.2 Constructor & Destructor Documentation . . . 1022
8.128.2.1 print_csrc_double() . . . 1022
8.129 GiNaC::print_csrc_float Class Reference . . . 1023
8.129.1 Detailed Description . . . 1024
8.129.2 Constructor & Destructor Documentation . . . 1024
8.129.2.1 print_csrc_float() . . . 1024
8.130 GiNaC::print_dflt Class Reference . . . 1025
8.130.1 Detailed Description . . . 1026
8.130.2 Constructor & Destructor Documentation . . . 1026
8.130.2.1 print_dflt() . . . 1026
8.131 GiNaC::print_functor Class Reference . . . 1026
8.131.1 Detailed Description . . . 1027
8.131.2 Constructor & Destructor Documentation . . . 1027
8.131.2.1 print_functor() [1/5] . . . 1027
8.131.2.2 print_functor() [2/5] . . . 1027
8.131.2.3 print_functor() [3/5] . . . 1027
8.131.2.4 print_functor() [4/5] . . . 1027
8.131.2.5 print_functor() [5/5] . . . 1028
8.131.3 Member Function Documentation . . . 1028

8.131.3.1 operator=() . . . 1028
8.131.3.2 operator()() . . . 1028
8.131.3.3 is_valid() . . . 1028
8.131.4 Member Data Documentation . . . 1028
8.131.4.1 impl . . . 1028
8.132 GiNaC::print_functor_impl Class Reference . . . 1029
8.132.1 Detailed Description . . . 1029
8.132.2 Constructor & Destructor Documentation . . . 1029
8.132.2.1 ∼print_functor_impl() . . . 1029
8.132.3 Member Function Documentation . . . 1029
8.132.3.1 duplicate() . . . 1029
8.132.3.2 operator()() . . . 1030
8.133 GiNaC::print_latex Class Reference . . . 1030
8.133.1 Detailed Description . . . 1031
8.133.2 Constructor & Destructor Documentation . . . 1031
8.133.2.1 print_latex() . . . 1031
8.134 GiNaC::print_memfun_handler< T, C > Class Template Reference . . . 1032
8.134.1 Detailed Description . . . 1033
8.134.2 Member Typedef Documentation . . . 1033
8.134.2.1 F . . . 1033
8.134.3 Constructor & Destructor Documentation . . . 1033
8.134.3.1 print_memfun_handler() . . . 1033
8.134.4 Member Function Documentation . . . 1033
8.134.4.1 duplicate() . . . 1033
8.134.4.2 operator()() . . . 1034
8.134.5 Member Data Documentation . . . 1034
8.134.5.1 f . . . 1034
8.135 GiNaC::print_options Class Reference . . . 1034
8.135.1 Detailed Description . . . 1034
8.135.2 Member Enumeration Documentation . . . 1035
8.135.2.1 anonymous enum . . . 1035
8.136 GiNaC::print_ptrfun_handler< T, C > Class Template Reference . . . 1035
8.136.1 Detailed Description . . . 1036
8.136.2 Member Typedef Documentation . . . 1036
8.136.2.1 F . . . 1036
8.136.3 Constructor & Destructor Documentation . . . 1036
8.136.3.1 print_ptrfun_handler() . . . 1036
8.136.4 Member Function Documentation . . . 1037
8.136.4.1 duplicate() . . . 1037
8.136.4.2 operator()() . . . 1037
8.136.5 Member Data Documentation . . . 1037
8.136.5.1 f . . . 1037

8.137 GiNaC::print_python Class Reference . . . 1038
8.137.1 Detailed Description . . . 1039
8.137.2 Constructor & Destructor Documentation . . . 1039
8.137.2.1 print_python() . . . 1039
8.138 GiNaC::print_python_repr Class Reference . . . 1039
8.138.1 Detailed Description . . . 1040
8.138.2 Constructor & Destructor Documentation . . . 1040
8.138.2.1 print_python_repr() . . . 1040
8.139 GiNaC::print_tree Class Reference . . . 1041
8.139.1 Detailed Description . . . 1042
8.139.2 Constructor & Destructor Documentation . . . 1042
8.139.2.1 print_tree() [1/2] . . . 1042
8.139.2.2 print_tree() [2/2] . . . 1042
8.139.3 Member Data Documentation . . . 1042
8.139.3.1 delta_indent . . . 1042
8.140 GiNaC::archive_node::property Struct Reference . . . 1043
8.140.1 Detailed Description . . . 1043
8.140.2 Constructor & Destructor Documentation . . . 1043
8.140.2.1 property() [1/2] . . . 1043
8.140.2.2 property() [2/2] . . . 1043
8.140.3 Member Data Documentation . . . 1044
8.140.3.1 type . . . 1044
8.140.3.2 name . . . 1044
8.140.3.3 value . . . 1044
8.141 GiNaC::archive_node::property_info Struct Reference . . . 1044
8.141.1 Detailed Description . . . 1045
8.141.2 Constructor & Destructor Documentation . . . 1045
8.141.2.1 property_info() [1/2] . . . 1045
8.141.2.2 property_info() [2/2] . . . 1045
8.141.3 Member Data Documentation . . . 1045
8.141.3.1 type . . . 1045
8.141.3.2 name . . . 1045
8.141.3.3 count . . . 1046
8.142 GiNaC::pseries Class Reference . . . 1046
8.142.1 Detailed Description . . . 1051
8.142.2 Constructor & Destructor Documentation . . . 1051
8.142.2.1 pseries() [1/2] . . . 1051
8.142.2.2 pseries() [2/2] . . . 1052
8.142.3 Member Function Documentation . . . 1052
8.142.3.1 precedence() . . . 1052
8.142.3.2 nops() . . . 1052
8.142.3.3 op() . . . 1053

8.142.3.4 degree() . . . 1053
8.142.3.5 ldegree() . . . 1053
8.142.3.6 coeff() . . . 1053
8.142.3.7 collect() . . . 1054
8.142.3.8 eval() . . . 1054
8.142.3.9 evalf() . . . 1054
8.142.3.10 series() . . . 1054
8.142.3.11 subs() . . . 1055
8.142.3.12 normal() . . . 1055
8.142.3.13 expand() . . . 1055
8.142.3.14 conjugate() . . . 1056
8.142.3.15 real_part() . . . 1056
8.142.3.16 imag_part() . . . 1056
8.142.3.17 eval_integ() . . . 1056
8.142.3.18 evalm() . . . 1056
8.142.3.19 archive() . . . 1057
8.142.3.20 read_archive() . . . 1057
8.142.3.21 derivative() . . . 1057
8.142.3.22 get_var() . . . 1057
8.142.3.23 get_point() . . . 1058
8.142.3.24 convert_to_poly() . . . 1058
8.142.3.25 is_compatible_to() . . . 1058
8.142.3.26 is_zero() . . . 1058
8.142.3.27 is_terminating() . . . 1058
8.142.3.28 coeffop() . . . 1059
8.142.3.29 exponop() . . . 1059
8.142.3.30 add_series() . . . 1059
8.142.3.31 mul_const() . . . 1059
8.142.3.32 mul_series() . . . 1060
8.142.3.33 power_const() . . . 1060
8.142.3.34 shift_exponents() . . . 1060
8.142.3.35 print_series() . . . 1061
8.142.3.36 do_print() . . . 1061
8.142.3.37 do_print_latex() . . . 1061
8.142.3.38 do_print_tree() . . . 1061
8.142.3.39 do_print_python() . . . 1062
8.142.3.40 do_print_python_repr() . . . 1062
8.142.4 Member Data Documentation . . . 1062
8.142.4.1 seq . . . 1062
8.142.4.2 var . . . 1062
8.142.4.3 point . . . 1063
8.143 GiNaC::psi1_SERIAL Class Reference . . . 1063

8.143.1 Detailed Description . . . 1063
8.143.2 Member Data Documentation . . . 1063
8.143.2.1 serial . . . 1063
8.144 GiNaC::psi2_SERIAL Class Reference . . . 1064
8.144.1 Detailed Description . . . 1064
8.144.2 Member Data Documentation . . . 1064
8.144.2.1 serial . . . 1064
8.145 GiNaC::ptr< T > Class Template Reference . . . 1064
8.145.1 Detailed Description . . . 1065
8.145.2 Constructor & Destructor Documentation . . . 1066
8.145.2.1 ptr() [1/3] . . . 1066
8.145.2.2 ptr() [2/3] . . . 1066
8.145.2.3 ptr() [3/3] . . . 1066
8.145.2.4 ~ptr() . . . 1066
8.145.3 Member Function Documentation . . . 1066
8.145.3.1 operator=() . . . 1066
8.145.3.2 operator*() . . . 1067
8.145.3.3 operator->() . . . 1067
8.145.3.4 makewritable() . . . 1067
8.145.3.5 swap() . . . 1067
8.145.3.6 operator==() . . . 1067
8.145.3.7 operator"!=() . . . 1067
8.145.4 Friends And Related Symbol Documentation . . . 1068
8.145.4.1 std::less< ptr< T > > . . . 1068
8.145.4.2 get_pointer . . . 1068
8.145.4.3 operator== [1/2] . . . 1068
8.145.4.4 operator"!= [1/2] . . . 1068
8.145.4.5 operator== [2/2] . . . 1068
8.145.4.6 operator"!= [2/2] . . . 1069
8.145.4.7 operator<< . . . 1069
8.145.5 Member Data Documentation . . . 1069
8.145.5.1 p . . . 1069
8.146 GiNaC::realsymbol Class Reference . . . 1069
8.146.1 Detailed Description . . . 1074
8.146.2 Constructor & Destructor Documentation . . . 1074
8.146.2.1 realsymbol() [1/3] . . . 1074
8.146.2.2 realsymbol() [2/3] . . . 1075
8.146.2.3 realsymbol() [3/3] . . . 1075
8.146.3 Member Function Documentation . . . 1075
8.146.3.1 get_domain() . . . 1075
8.146.3.2 conjugate() . . . 1075
8.146.3.3 real_part() . . . 1075

8.146.3.4 imag_part() . . . 1075
8.146.3.5 duplicate() . . . 1076
8.147 GiNaC::refcounted Class Reference . . . 1076
8.147.1 Detailed Description . . . 1078
8.147.2 Constructor & Destructor Documentation . . . 1078
8.147.2.1 refcounted() . . . 1078
8.147.3 Member Function Documentation . . . 1078
8.147.3.1 add_reference() . . . 1078
8.147.3.2 remove_reference() . . . 1078
8.147.3.3 get_refcount() . . . 1078
8.147.3.4 set_refcount() . . . 1078
8.147.4 Member Data Documentation . . . 1079
8.147.4.1 refcount . . . 1079
8.148 GiNaC::registered_class_options Class Reference . . . 1079
8.148.1 Detailed Description . . . 1079
8.148.2 Constructor & Destructor Documentation . . . 1080
8.148.2.1 registered_class_options() . . . 1080
8.148.3 Member Function Documentation . . . 1080
8.148.3.1 get_name() . . . 1080
8.148.3.2 get_parent_name() . . . 1080
8.148.3.3 get_id() . . . 1080
8.148.3.4 get_print_dispatch_table() . . . 1080
8.148.3.5 print_func() [1/3] . . . 1080
8.148.3.6 print_func() [2/3] . . . 1081
8.148.3.7 print_func() [3/3] . . . 1081
8.148.3.8 set_print_func() . . . 1081
8.148.4 Member Data Documentation . . . 1081
8.148.4.1 name . . . 1081
8.148.4.2 parent_name . . . 1081
8.148.4.3 tinfo_key . . . 1081
8.148.4.4 print_dispatch_table . . . 1082
8.149 GiNaC::relational Class Reference . . . 1082
8.149.1 Detailed Description . . . 1087
8.149.2 Member Typedef Documentation . . . 1087
8.149.2.1 safe_bool . . . 1087
8.149.3 Member Enumeration Documentation . . . 1087
8.149.3.1 operators . . . 1087
8.149.4 Constructor & Destructor Documentation . . . 1088
8.149.4.1 relational() . . . 1088
8.149.5 Member Function Documentation . . . 1088
8.149.5.1 precedence() . . . 1088
8.149.5.2 info() . . . 1088

8.149.5.3 nops() . . . 1089
8.149.5.4 op() . . . 1089
8.149.5.5 map() . . . 1089
8.149.5.6 subs() . . . 1089
8.149.5.7 archive() . . . 1089
8.149.5.8 read_archive() . . . 1090
8.149.5.9 canonical() . . . 1090
8.149.5.10 eval_ncmul() . . . 1090
8.149.5.11 match_same_type() . . . 1090
8.149.5.12 return_type() . . . 1091
8.149.5.13 return_type_tinfo() . . . 1091
8.149.5.14 calchash() . . . 1091
8.149.5.15 do_print() . . . 1091
8.149.5.16 do_print_python_repr() . . . 1091
8.149.5.17 lhs() . . . 1092
8.149.5.18 rhs() . . . 1092
8.149.5.19 make_safe_bool() . . . 1092
8.149.5.20 operator safe_bool() . . . 1092
8.149.5.21 operator"!() . . . 1092
8.149.6 Member Data Documentation . . . 1093
8.149.6.1 lh . . . 1093
8.149.6.2 rh . . . 1093
8.149.6.3 o . . . 1093
8.150 GiNaC::remember_strategies Class Reference . . . 1093
8.150.1 Detailed Description . . . 1093
8.150.2 Member Enumeration Documentation . . . 1094
8.150.2.1 anonymous enum . . . 1094
8.151 GiNaC::remember_table Class Reference . . . 1094
8.151.1 Detailed Description . . . 1095
8.151.2 Constructor & Destructor Documentation . . . 1096
8.151.2.1 remember_table() [1/2] . . . 1096
8.151.2.2 remember_table() [2/2] . . . 1096
8.151.3 Member Function Documentation . . . 1096
8.151.3.1 lookup_entry() . . . 1096
8.151.3.2 add_entry() . . . 1096
8.151.3.3 clear_all_entries() . . . 1096
8.151.3.4 show_statistics() . . . 1096
8.151.3.5 remember_tables() . . . 1097
8.151.3.6 init_table() . . . 1097
8.151.4 Member Data Documentation . . . 1097
8.151.4.1 table_size . . . 1097
8.151.4.2 max_assoc_size . . . 1097

8.151.4.3 remember_strategy . . . 1097
8.152 GiNaC::remember_table_entry Class Reference . . . 1098
8.152.1 Detailed Description . . . 1099
8.152.2 Constructor & Destructor Documentation . . . 1099
8.152.2.1 remember_table_entry() . . . 1099
8.152.3 Member Function Documentation . . . 1099
8.152.3.1 is_equal() . . . 1099
8.152.3.2 get_result() . . . 1099
8.152.3.3 get_last_access() . . . 1099
8.152.3.4 get_successful_hits() . . . 1100
8.152.4 Member Data Documentation . . . 1100
8.152.4.1 hashvalue . . . 1100
8.152.4.2 seq . . . 1100
8.152.4.3 result . . . 1100
8.152.4.4 last_access . . . 1100
8.152.4.5 successful_hits . . . 1100
8.152.4.6 access_counter . . . 1100
8.153 GiNaC::remember_table_list Class Reference . . . 1101
8.153.1 Detailed Description . . . 1102
8.153.2 Constructor & Destructor Documentation . . . 1102
8.153.2.1 remember_table_list() . . . 1102
8.153.3 Member Function Documentation . . . 1102
8.153.3.1 add_entry() . . . 1102
8.153.3.2 lookup_entry() . . . 1102
8.153.4 Member Data Documentation . . . 1102
8.153.4.1 max_assoc_size . . . 1102
8.153.4.2 remember_strategy . . . 1103
8.154 GiNaC::return_type_t Struct Reference . . . 1103
8.154.1 Detailed Description . . . 1103
8.154.2 Member Function Documentation . . . 1103
8.154.2.1 operator<() . . . 1103
8.154.2.2 operator==() . . . 1104
8.154.2.3 operator"!=() . . . 1104
8.154.3 Member Data Documentation . . . 1104
8.154.3.1 tinfo . . . 1104
8.154.3.2 rl . . . 1104
8.155 GiNaC::return_types Class Reference . . . 1104
8.155.1 Member Enumeration Documentation . . . 1105
8.155.1.1 anonymous enum . . . 1105
8.156 GiNaC::relational::safe_bool_helper Struct Reference . . . 1105
8.156.1 Member Function Documentation . . . 1105
8.156.1.1 nonnull() . . . 1105

8.157 GiNaC::scalar_products Class Reference . . . 1105
8.157.1 Detailed Description . . . 1106
8.157.2 Member Function Documentation . . . 1106
8.157.2.1 add() [1/2] . . . 1106
8.157.2.2 add() [2/2] . . . 1106
8.157.2.3 add_vectors() . . . 1107
8.157.2.4 clear() . . . 1107
8.157.2.5 is_defined() . . . 1107
8.157.2.6 evaluate() . . . 1107
8.157.2.7 debugprint() . . . 1107
8.157.3 Member Data Documentation . . . 1108
8.157.3.1 spm . . . 1108
8.158 GiNaC::series_options Class Reference . . . 1108
8.158.1 Detailed Description . . . 1108
8.158.2 Member Enumeration Documentation . . . 1108
8.158.2.1 anonymous enum . . . 1108
8.159 GiNaC::solve_algo Class Reference . . . 1109
8.159.1 Detailed Description . . . 1109
8.159.2 Member Enumeration Documentation . . . 1109
8.159.2.1 anonymous enum . . . 1109
8.160 GiNaC::spinidx Class Reference . . . 1110
8.160.1 Detailed Description . . . 1116
8.160.2 Constructor & Destructor Documentation . . . 1116
8.160.2.1 spinidx() . . . 1116
8.160.3 Member Function Documentation . . . 1117
8.160.3.1 is_dummy_pair_same_type() . . . 1117
8.160.3.2 conjugate() . . . 1117
8.160.3.3 archive() . . . 1117
8.160.3.4 read_archive() . . . 1118
8.160.3.5 match_same_type() . . . 1118
8.160.3.6 is_dotted() . . . 1118
8.160.3.7 is_undotted() . . . 1119
8.160.3.8 toggle_dot() . . . 1119
8.160.3.9 toggle_variance_dot() . . . 1119
8.160.3.10 do_print() . . . 1119
8.160.3.11 do_print_latex() . . . 1119
8.160.3.12 do_print_tree() . . . 1119
8.160.4 Member Data Documentation . . . 1120
8.160.4.1 dotted . . . 1120
8.161 GiNaC::spinmetric Class Reference . . . 1120
8.161.1 Detailed Description . . . 1125
8.161.2 Member Function Documentation . . . 1125

8.161.2.1 info() . . . 1125
8.161.2.2 eval_indexed() . . . 1125
8.161.2.3 contract_with() . . . 1126
8.161.2.4 do_print() . . . 1126
8.161.2.5 do_print_latex() . . . 1126
8.162 GiNaC::spmapkey Class Reference . . . 1126
8.162.1 Constructor & Destructor Documentation . . . 1128
8.162.1.1 spmapkey() [1/2] . . . 1128
8.162.1.2 spmapkey() [2/2] . . . 1128
8.162.2 Member Function Documentation . . . 1128
8.162.2.1 operator==() . . . 1128
8.162.2.2 operator<() . . . 1128
8.162.2.3 debugprint() . . . 1128
8.162.3 Member Data Documentation . . . 1129
8.162.3.1 v1 . . . 1129
8.162.3.2 v2 . . . 1129
8.162.3.3 dim . . . 1129
8.163 GiNaC::status_flags Class Reference . . . 1129
8.163.1 Detailed Description . . . 1129
8.163.2 Member Enumeration Documentation . . . 1130
8.163.2.1 anonymous enum . . . 1130
8.164 GiNaC::structure< T, ComparisonPolicy > Class Template Reference . . . 1130
8.164.1 Detailed Description . . . 1135
8.164.2 Constructor & Destructor Documentation . . . 1135
8.164.2.1 structure() . . . 1135
8.164.3 Member Function Documentation . . . 1135
8.164.3.1 get_class_name() . . . 1135
8.164.3.2 eval() . . . 1136
8.164.3.3 evalm() . . . 1136
8.164.3.4 eval_ncmul() . . . 1136
8.164.3.5 eval_indexed() . . . 1136
8.164.3.6 print() . . . 1137
8.164.3.7 precedence() . . . 1137
8.164.3.8 info() . . . 1137
8.164.3.9 nops() . . . 1137
8.164.3.10 op() . . . 1138
8.164.3.11 operator[]() [1/4] . . . 1138
8.164.3.12 operator[]() [2/4] . . . 1138
8.164.3.13 let_op() . . . 1138
8.164.3.14 operator[]() [3/4] . . . 1138
8.164.3.15 operator[]() [4/4] . . . 1139
8.164.3.16 has() . . . 1139

8.164.3.17 match() . . . 1139
8.164.3.18 match_same_type() . . . 1139
8.164.3.19 subs() . . . 1140
8.164.3.20 map() . . . 1140
8.164.3.21 degree() . . . 1140
8.164.3.22 ldegree() . . . 1140
8.164.3.23 coeff() . . . 1141
8.164.3.24 expand() . . . 1141
8.164.3.25 collect() . . . 1141
8.164.3.26 derivative() . . . 1142
8.164.3.27 series() . . . 1142
8.164.3.28 normal() . . . 1142
8.164.3.29 to_rational() . . . 1143
8.164.3.30 to_polynomial() . . . 1143
8.164.3.31 integer_content() . . . 1143
8.164.3.32 smod() . . . 1143
8.164.3.33 max_coefficient() . . . 1144
8.164.3.34 get_free_indices() . . . 1144
8.164.3.35 add_indexed() . . . 1144
8.164.3.36 scalar_mul_indexed() . . . 1145
8.164.3.37 contract_with() . . . 1145
8.164.3.38 return_type() . . . 1146
8.164.3.39 return_type_tinfo() . . . 1146
8.164.3.40 is_equal_same_type() . . . 1146
8.164.3.41 calchash() . . . 1146
8.164.3.42 operator->() . . . 1147
8.164.3.43 get_struct() [1/2] . . . 1147
8.164.3.44 get_struct() [2/2] . . . 1147
8.164.4 Member Data Documentation . . . 1147
8.164.4.1 obj . . . 1147
8.165 GiNaC::su3d Class Reference . . . 1148
8.165.1 Detailed Description . . . 1152
8.165.2 Member Function Documentation . . . 1152
8.165.2.1 eval_indexed() . . . 1152
8.165.2.2 contract_with() . . . 1152
8.165.2.3 return_type() . . . 1153
8.165.2.4 do_print() . . . 1153
8.165.2.5 do_print_latex() . . . 1153
8.166 GiNaC::su3f Class Reference . . . 1153
8.166.1 Detailed Description . . . 1157
8.166.2 Member Function Documentation . . . 1158
8.166.2.1 eval_indexed() . . . 1158

8.166.2.2 contract_with() . . . 1158
8.166.2.3 return_type() . . . 1158
8.166.2.4 do_print() . . . 1158
8.166.2.5 do_print_latex() . . . 1158
8.167 GiNaC::su3one Class Reference . . . 1159
8.167.1 Detailed Description . . . 1163
8.167.2 Member Function Documentation . . . 1163
8.167.2.1 do_print() . . . 1163
8.167.2.2 do_print_latex() . . . 1163
8.168 GiNaC::su3t Class Reference . . . 1164
8.168.1 Detailed Description . . . 1168
8.168.2 Member Function Documentation . . . 1168
8.168.2.1 contract_with() . . . 1168
8.168.2.2 do_print() . . . 1168
8.168.2.3 do_print_latex() . . . 1169
8.169 GiNaC::subs_options Class Reference . . . 1169
8.169.1 Detailed Description . . . 1169
8.169.2 Member Enumeration Documentation . . . 1169
8.169.2.1 anonymous enum . . . 1169
8.170 GiNaC::sy_is_less Class Reference . . . 1170
8.170.1 Constructor & Destructor Documentation . . . 1170
8.170.1.1 sy_is_less() . . . 1170
8.170.2 Member Function Documentation . . . 1170
8.170.2.1 operator()() . . . 1170
8.170.3 Member Data Documentation . . . 1170
8.170.3.1 v . . . 1170
8.171 GiNaC::sy_swap Class Reference . . . 1171
8.171.1 Constructor & Destructor Documentation . . . 1171
8.171.1.1 sy_swap() . . . 1171
8.171.2 Member Function Documentation . . . 1171
8.171.2.1 operator()() . . . 1171
8.171.3 Member Data Documentation . . . 1171
8.171.3.1 v . . . 1171
8.171.3.2 swapped . . . 1172
8.172 GiNaC::sym_desc Struct Reference . . . 1172
8.172.1 Detailed Description . . . 1173
8.172.2 Constructor & Destructor Documentation . . . 1173
8.172.2.1 sym_desc() . . . 1173
8.172.3 Member Function Documentation . . . 1174
8.172.3.1 operator<() . . . 1174
8.172.4 Member Data Documentation . . . 1174
8.172.4.1 sym . . . 1174

8.172.4.2 deg_a . . . 1174
8.172.4.3 deg_b . . . 1174
8.172.4.4 ldeg_a . . . 1174
8.172.4.5 ldeg_b . . . 1175
8.172.4.6 max_deg . . . 1175
8.172.4.7 max_lcnops . . . 1175
8.173 GiNaC::symbol Class Reference . . . 1175
8.173.1 Detailed Description . . . 1180
8.173.2 Constructor & Destructor Documentation . . . 1180
8.173.2.1 symbol() [1/2] . . . 1180
8.173.2.2 symbol() [2/2] . . . 1180
8.173.3 Member Function Documentation . . . 1181
8.173.3.1 info() . . . 1181
8.173.3.2 eval() . . . 1181
8.173.3.3 evalf() . . . 1181
8.173.3.4 series() . . . 1181
8.173.3.5 subs() . . . 1182
8.173.3.6 normal() . . . 1182
8.173.3.7 to_rational() . . . 1182
8.173.3.8 to_polynomial() . . . 1182
8.173.3.9 conjugate() . . . 1183
8.173.3.10 real_part() . . . 1183
8.173.3.11 imag_part() . . . 1183
8.173.3.12 is_polynomial() . . . 1183
8.173.3.13 archive() . . . 1183
8.173.3.14 read_archive() . . . 1184
8.173.3.15 derivative() . . . 1184
8.173.3.16 is_equal_same_type() . . . 1184
8.173.3.17 calchash() . . . 1185
8.173.3.18 get_domain() . . . 1185
8.173.3.19 set_name() . . . 1185
8.173.3.20 set_TeX_name() . . . 1185
8.173.3.21 get_name() . . . 1185
8.173.3.22 get_TeX_name() . . . 1186
8.173.3.23 do_print() . . . 1186
8.173.3.24 do_print_latex() . . . 1186
8.173.3.25 do_print_tree() . . . 1186
8.173.3.26 do_print_python_repr() . . . 1186
8.173.4 Member Data Documentation . . . 1187
8.173.4.1 serial . . . 1187
8.173.4.2 name . . . 1187
8.173.4.3 TeX_name . . . 1187

8.173.4.4 next_serial . . . 1187
8.174 GiNaC::symbolset Class Reference . . . 1187
8.174.1 Constructor & Destructor Documentation . . . 1188
8.174.1.1 symbolset() . . . 1188
8.174.2 Member Function Documentation . . . 1188
8.174.2.1 insert_symbols() . . . 1188
8.174.2.2 has() . . . 1188
8.174.3 Member Data Documentation . . . 1188
8.174.3.1 s . . . 1188
8.175 GiNaC::symmetry Class Reference . . . 1189
8.175.1 Detailed Description . . . 1194
8.175.2 Member Enumeration Documentation . . . 1194
8.175.2.1 symmetry_type . . . 1194
8.175.3 Constructor & Destructor Documentation . . . 1194
8.175.3.1 symmetry() [1/2] . . . 1194
8.175.3.2 symmetry() [2/2] . . . 1194
8.175.4 Member Function Documentation . . . 1195
8.175.4.1 archive() . . . 1195
8.175.4.2 read_archive() . . . 1195
8.175.4.3 calchash() . . . 1195
8.175.4.4 get_type() . . . 1196
8.175.4.5 set_type() . . . 1196
8.175.4.6 add() . . . 1196
8.175.4.7 validate() . . . 1196
8.175.4.8 has_symmetry() . . . 1196
8.175.4.9 has_nonsymmetric() . . . 1197
8.175.4.10 has_cyclic() . . . 1197
8.175.4.11 do_print() . . . 1197
8.175.4.12 do_print_tree() . . . 1197
8.175.5 Friends And Related Symbol Documentation . . . 1197
8.175.5.1 sy_is_less . . . 1197
8.175.5.2 sy_swap . . . 1198
8.175.5.3 canonicalize . . . 1198
8.175.6 Member Data Documentation . . . 1198
8.175.6.1 type . . . 1198
8.175.6.2 indices . . . 1198
8.175.6.3 children . . . 1199
8.176 GiNaC::symminfo Class Reference . . . 1199
8.176.1 Detailed Description . . . 1200
8.176.2 Constructor & Destructor Documentation . . . 1200
8.176.2.1 symminfo() [1/2] . . . 1200
8.176.2.2 symminfo() [2/2] . . . 1200

8.176.3 Member Data Documentation . . . 1200
8.176.3.1 symmterm . . . 1200
8.176.3.2 coeff . . . 1201
8.176.3.3 orig . . . 1201
8.176.3.4 num . . . 1201
8.177 GiNaC::symminfo_is_less_by_orig Class Reference . . . 1201
8.177.1 Member Function Documentation . . . 1201
8.177.1.1 operator()() . . . 1201
8.178 GiNaC::symminfo_is_less_by_symmterm Class Reference . . . 1202
8.178.1 Member Function Documentation . . . 1202
8.178.1.1 operator()() . . . 1202
8.179 GiNaC::tensdelta Class Reference . . . 1202
8.179.1 Detailed Description . . . 1206
8.179.2 Member Function Documentation . . . 1207
8.179.2.1 info() . . . 1207
8.179.2.2 eval_indexed() . . . 1207
8.179.2.3 contract_with() . . . 1207
8.179.2.4 return_type() . . . 1207
8.179.2.5 do_print() . . . 1208
8.179.2.6 do_print_latex() . . . 1208
8.180 GiNaC::tensepsilon Class Reference . . . 1208
8.180.1 Detailed Description . . . 1213
8.180.2 Constructor & Destructor Documentation . . . 1213
8.180.2.1 tensepsilon() . . . 1213
8.180.3 Member Function Documentation . . . 1213
8.180.3.1 info() . . . 1213
8.180.3.2 eval_indexed() . . . 1213
8.180.3.3 contract_with() . . . 1214
8.180.3.4 archive() . . . 1214
8.180.3.5 read_archive() . . . 1214
8.180.3.6 return_type() . . . 1214
8.180.3.7 do_print() . . . 1215
8.180.3.8 do_print_latex() . . . 1215
8.180.4 Member Data Documentation . . . 1215
8.180.4.1 minkowski . . . 1215
8.180.4.2 pos_sig . . . 1215
8.181 GiNaC::tensmetric Class Reference . . . 1216
8.181.1 Detailed Description . . . 1220
8.181.2 Member Function Documentation . . . 1221
8.181.2.1 info() . . . 1221
8.181.2.2 eval_indexed() . . . 1221
8.181.2.3 contract_with() . . . 1221

8.181.2.4 return_type() . . . 1221
8.181.2.5 do_print() . . . 1222
8.182 GiNaC::tensor Class Reference . . . 1222
8.182.1 Detailed Description . . . 1226
8.182.2 Member Function Documentation . . . 1226
8.182.2.1 return_type() . . . 1226
8.182.2.2 replace_contr_index() . . . 1227
8.183 GiNaC::terminfo Class Reference . . . 1227
8.183.1 Detailed Description . . . 1228
8.183.2 Constructor & Destructor Documentation . . . 1228
8.183.2.1 terminfo() . . . 1228
8.183.3 Member Data Documentation . . . 1228
8.183.3.1 orig . . . 1228
8.183.3.2 symm . . . 1228
8.184 GiNaC::terminfo_is_less Class Reference . . . 1229
8.184.1 Member Function Documentation . . . 1229
8.184.1.1 operator()() . . . 1229
8.185 GiNaC::Gt_detail::TransformExpressionWithCache< Type > Class Template Reference . . . 1229
8.185.1 Constructor & Destructor Documentation . . . 1230
8.185.1.1 TransformExpressionWithCache() . . . 1230
8.185.2 Member Function Documentation . . . 1230
8.185.2.1 operator()() . . . 1230
8.185.2.2 print_cache() . . . 1231
8.185.2.3 get_cache() . . . 1231
8.185.2.4 clear_cache() . . . 1231
8.185.2.5 check_type() [1/2] . . . 1231
8.185.2.6 check_type() [2/2] . . . 1231
8.185.2.7 impl() . . . 1231
8.185.3 Member Data Documentation . . . 1232
8.185.3.1 func_obj . . . 1232
8.185.3.2 func_default . . . 1232
8.185.3.3 cache . . . 1232
8.186 GiNaC::class_info< OPT >::tree_node Struct Reference . . . 1232
8.186.1 Constructor & Destructor Documentation . . . 1233
8.186.1.1 tree_node() . . . 1233
8.186.2 Member Function Documentation . . . 1233
8.186.2.1 add_child() . . . 1233
8.186.3 Member Data Documentation . . . 1233
8.186.3.1 children . . . 1233
8.186.3.2 info . . . 1233
8.187 GiNaC::unarchive_table_t Class Reference . . . 1234
8.187.1 Constructor & Destructor Documentation . . . 1234

8.187.1.1 unarchive_table_t() . . . 1234
8.187.1.2 ~unarchive_table_t() . . . 1234
8.187.2 Member Function Documentation . . . 1234
8.187.2.1 find() . . . 1234
8.187.2.2 insert() . . . 1234
8.187.3 Member Data Documentation . . . 1235
8.187.3.1 usecount . . . 1235
8.187.3.2 unarch_map . . . 1235
8.188 GiNaC::user_defined_kernel Class Reference . . . 1235
8.188.1 Detailed Description . . . 1240
8.188.2 Constructor & Destructor Documentation . . . 1241
8.188.2.1 user_defined_kernel() . . . 1241
8.188.3 Member Function Documentation . . . 1241
8.188.3.1 nops() . . . 1241
8.188.3.2 op() . . . 1241
8.188.3.3 let_op() . . . 1241
8.188.3.4 is_numeric() . . . 1241
8.188.3.5 Laurent_series() . . . 1242
8.188.3.6 uses_Laurent_series() . . . 1242
8.188.3.7 do_print() . . . 1242
8.188.4 Member Data Documentation . . . 1242
8.188.4.1 f . . . 1242
8.188.4.2 x . . . 1242
8.189 GiNaC::varidx Class Reference . . . 1243
8.189.1 Detailed Description . . . 1249
8.189.2 Constructor & Destructor Documentation . . . 1249
8.189.2.1 varidx() . . . 1249
8.189.3 Member Function Documentation . . . 1249
8.189.3.1 is_dummy_pair_same_type() . . . 1249
8.189.3.2 archive() . . . 1249
8.189.3.3 read_archive() . . . 1250
8.189.3.4 match_same_type() . . . 1250
8.189.3.5 is_covariant() . . . 1250
8.189.3.6 is_contravariant() . . . 1251
8.189.3.7 toggle_variance() . . . 1251
8.189.3.8 do_print() . . . 1251
8.189.3.9 do_print_tree() . . . 1251
8.189.4 Member Data Documentation . . . 1251
8.189.4.1 covariant . . . 1251
8.190 GiNaC::visitor Class Reference . . . 1252
8.190.1 Detailed Description . . . 1252
8.190.2 Constructor & Destructor Documentation . . . 1252

8.190.2.1 ~visitor() . . . 1252
8.191 GiNaC::wildcard Class Reference . . . 1252
8.191.1 Detailed Description . . . 1256
8.191.2 Constructor & Destructor Documentation . . . 1256
8.191.2.1 wildcard() . . . 1256
8.191.3 Member Function Documentation . . . 1257
8.191.3.1 match() . . . 1257
8.191.3.2 archive() . . . 1257
8.191.3.3 read_archive() . . . 1257
8.191.3.4 calchash() . . . 1257
8.191.3.5 get_label() . . . 1258
8.191.3.6 do_print() . . . 1258
8.191.3.7 do_print_tree() . . . 1258
8.191.3.8 do_print_python_repr() . . . 1258
8.191.4 Member Data Documentation . . . 1258
8.191.4.1 label . . . 1258
8.192 GiNaC::zeta1_SERIAL Class Reference . . . 1259
8.192.1 Detailed Description . . . 1259
8.192.2 Member Data Documentation . . . 1259
8.192.2.1 serial . . . 1259
8.193 GiNaC::zeta2_SERIAL Class Reference . . . 1259
8.193.1 Detailed Description . . . 1260
8.193.2 Member Data Documentation . . . 1260
8.193.2.1 serial . . . 1260

9 File Documentation **1261**
9.1 add.cpp File Reference . . . 1261
9.1.1 Detailed Description . . . 1261
9.2 add.h File Reference . . . 1262
9.2.1 Detailed Description . . . 1262
9.3 archive.cpp File Reference . . . 1262
9.3.1 Detailed Description . . . 1263
9.4 archive.h File Reference . . . 1263
9.4.1 Detailed Description . . . 1264
9.4.2 Macro Definition Documentation . . . 1264
9.4.2.1 GINAC_DECLARE_UNARCHIVER . . . 1264
9.4.2.2 GINAC_BIND_UNARCHIVER . . . 1265
9.5 assertion.h File Reference . . . 1265
9.5.1 Detailed Description . . . 1266
9.5.2 Macro Definition Documentation . . . 1266
9.5.2.1 GINAC_ASSERT . . . 1266
9.6 basic.cpp File Reference . . . 1267

9.6.1 Detailed Description . . . 1268
9.7 basic.h File Reference . . . 1268
9.7.1 Detailed Description . . . 1269
9.8 class_info.h File Reference . . . 1269
9.8.1 Detailed Description . . . 1270
9.9 clifford.cpp File Reference . . . 1270
9.9.1 Detailed Description . . . 1272
9.10 clifford.h File Reference . . . 1272
9.10.1 Detailed Description . . . 1274
9.11 color.cpp File Reference . . . 1274
9.11.1 Detailed Description . . . 1275
9.11.2 Macro Definition Documentation . . . 1275
9.11.2.1 TEST_PERMUTATION . . . 1275
9.11.2.2 CMPINDICES . . . 1276
9.12 color.h File Reference . . . 1276
9.12.1 Detailed Description . . . 1277
9.13 compiler.h File Reference . . . 1277
9.13.1 Detailed Description . . . 1277
9.13.2 Macro Definition Documentation . . . 1278
9.13.2.1 unlikely . . . 1278
9.13.2.2 likely . . . 1278
9.13.2.3 attribute_pure . . . 1278
9.14 constant.cpp File Reference . . . 1278
9.14.1 Detailed Description . . . 1279
9.15 constant.h File Reference . . . 1279
9.15.1 Detailed Description . . . 1279
9.16 container.h File Reference . . . 1280
9.16.1 Detailed Description . . . 1280
9.17 crc32.h File Reference . . . 1280
9.17.1 Detailed Description . . . 1281
9.18 ex.cpp File Reference . . . 1281
9.18.1 Detailed Description . . . 1281
9.19 ex.h File Reference . . . 1281
9.19.1 Detailed Description . . . 1283
9.20 excompiler.cpp File Reference . . . 1284
9.20.1 Detailed Description . . . 1284
9.21 excompiler.h File Reference . . . 1284
9.21.1 Detailed Description . . . 1285
9.22 expair.cpp File Reference . . . 1285
9.22.1 Detailed Description . . . 1286
9.23 expair.h File Reference . . . 1286
9.23.1 Detailed Description . . . 1286

9.24 expairseq.cpp File Reference . . . 1287
9.24.1 Detailed Description . . . 1287
9.25 expairseq.h File Reference . . . 1287
9.25.1 Detailed Description . . . 1288
9.26 exprseq.cpp File Reference . . . 1288
9.26.1 Detailed Description . . . 1288
9.27 exprseq.h File Reference . . . 1289
9.27.1 Detailed Description . . . 1289
9.28 factor.cpp File Reference . . . 1289
9.28.1 Detailed Description . . . 1290
9.28.2 Macro Definition Documentation . . . 1290
9.28.2.1 DCOUT . . . 1290
9.28.2.2 DCOUTVAR . . . 1290
9.28.2.3 DCOUT2 . . . 1290
9.28.2.4 USE_SAME_DEGREE_FACTOR . . . 1291
9.29 factor.h File Reference . . . 1291
9.29.1 Detailed Description . . . 1291
9.30 fail.cpp File Reference . . . 1291
9.30.1 Detailed Description . . . 1291
9.31 fail.h File Reference . . . 1292
9.31.1 Detailed Description . . . 1292
9.32 fderivative.cpp File Reference . . . 1292
9.32.1 Detailed Description . . . 1293
9.33 fderivative.h File Reference . . . 1293
9.33.1 Detailed Description . . . 1293
9.34 flags.h File Reference . . . 1293
9.34.1 Detailed Description . . . 1294
9.35 function.cpp File Reference . . . 1294
9.35.1 Detailed Description . . . 1295
9.36 function.h File Reference . . . 1295
9.36.1 Detailed Description . . . 1302
9.36.2 Macro Definition Documentation . . . 1302
9.36.2.1 DECLARE_FUNCTION_1P . . . 1302
9.36.2.2 DECLARE_FUNCTION_2P . . . 1302
9.36.2.3 DECLARE_FUNCTION_3P . . . 1302
9.36.2.4 DECLARE_FUNCTION_4P . . . 1303
9.36.2.5 DECLARE_FUNCTION_5P . . . 1303
9.36.2.6 DECLARE_FUNCTION_6P . . . 1303
9.36.2.7 DECLARE_FUNCTION_7P . . . 1303
9.36.2.8 DECLARE_FUNCTION_8P . . . 1304
9.36.2.9 DECLARE_FUNCTION_9P . . . 1304
9.36.2.10 DECLARE_FUNCTION_10P . . . 1304

9.36.2.11 DECLARE_FUNCTION_11P . . . 1304
9.36.2.12 DECLARE_FUNCTION_12P . . . 1305
9.36.2.13 DECLARE_FUNCTION_13P . . . 1305
9.36.2.14 DECLARE_FUNCTION_14P . . . 1305
9.36.2.15 REGISTER_FUNCTION . . . 1305
9.36.2.16 is_ex_the_function . . . 1306
9.37 ginac.h File Reference . . . 1306
9.37.1 Detailed Description . . . 1307
9.38 Gt.cpp File Reference . . . 1307
9.38.1 Detailed Description . . . 1307
9.38.2 Macro Definition Documentation . . . 1308
9.38.2.1 DEBUG_GT . . . 1308
9.39 Gt.h File Reference . . . 1308
9.39.1 Detailed Description . . . 1308
9.40 Gt_helpers.cpp File Reference . . . 1308
9.40.1 Detailed Description . . . 1309
9.41 Gt_helpers.h File Reference . . . 1309
9.41.1 Detailed Description . . . 1310
9.42 hash_map.h File Reference . . . 1310
9.42.1 Detailed Description . . . 1310
9.43 hash_seed.h File Reference . . . 1310
9.43.1 Detailed Description . . . 1311
9.44 idx.cpp File Reference . . . 1311
9.44.1 Detailed Description . . . 1312
9.45 idx.h File Reference . . . 1312
9.45.1 Detailed Description . . . 1313
9.46 indexed.cpp File Reference . . . 1313
9.46.1 Detailed Description . . . 1314
9.47 indexed.h File Reference . . . 1315
9.47.1 Detailed Description . . . 1315
9.48 inifcns.cpp File Reference . . . 1316
9.48.1 Detailed Description . . . 1319
9.49 inifcns.h File Reference . . . 1319
9.49.1 Detailed Description . . . 1320
9.50 inifcns_elliptic.cpp File Reference . . . 1320
9.50.1 Detailed Description . . . 1321
9.51 inifcns_gamma.cpp File Reference . . . 1322
9.51.1 Detailed Description . . . 1323
9.52 inifcns_nstdsums.cpp File Reference . . . 1323
9.52.1 Detailed Description . . . 1324
9.53 inifcns_trans.cpp File Reference . . . 1325
9.53.1 Detailed Description . . . 1328

9.54 integral.cpp File Reference . . . 1328
9.54.1 Detailed Description . . . 1329
9.55 integral.h File Reference . . . 1329
9.55.1 Detailed Description . . . 1329
9.56 integration_kernel.cpp File Reference . . . 1329
9.56.1 Detailed Description . . . 1331
9.57 integration_kernel.h File Reference . . . 1331
9.57.1 Detailed Description . . . 1332
9.58 lst.cpp File Reference . . . 1332
9.58.1 Detailed Description . . . 1333
9.59 lst.h File Reference . . . 1333
9.59.1 Detailed Description . . . 1333
9.60 matrix.cpp File Reference . . . 1333
9.60.1 Detailed Description . . . 1334
9.61 matrix.h File Reference . . . 1334
9.61.1 Detailed Description . . . 1335
9.62 mul.cpp File Reference . . . 1336
9.62.1 Detailed Description . . . 1336
9.63 mul.h File Reference . . . 1336
9.63.1 Detailed Description . . . 1337
9.64 ncmul.cpp File Reference . . . 1337
9.64.1 Detailed Description . . . 1338
9.65 ncmul.h File Reference . . . 1338
9.65.1 Detailed Description . . . 1338
9.66 normal.cpp File Reference . . . 1339
9.66.1 Detailed Description . . . 1341
9.66.2 Macro Definition Documentation . . . 1341
9.66.2.1 FAST_COMPARE . . . 1341
9.66.2.2 USE_REMEMBER . . . 1341
9.66.2.3 USE_TRIAL_DIVISION . . . 1341
9.66.2.4 STATISTICS . . . 1341
9.67 normal.h File Reference . . . 1341
9.67.1 Detailed Description . . . 1342
9.68 numeric.cpp File Reference . . . 1343
9.68.1 Detailed Description . . . 1346
9.69 numeric.h File Reference . . . 1346
9.69.1 Detailed Description . . . 1349
9.70 operators.cpp File Reference . . . 1349
9.70.1 Detailed Description . . . 1350
9.71 operators.h File Reference . . . 1351
9.71.1 Detailed Description . . . 1352
9.72 power.cpp File Reference . . . 1352

9.72.1 Detailed Description . . . 1353
9.73 power.h File Reference . . . 1353
9.73.1 Detailed Description . . . 1353
9.74 print.cpp File Reference . . . 1354
9.74.1 Detailed Description . . . 1354
9.75 print.h File Reference . . . 1354
9.75.1 Detailed Description . . . 1355
9.75.2 Macro Definition Documentation . . . 1356
9.75.2.1 GINAC_DECLARE_PRINT_CONTEXT_COMMON . . . 1356
9.75.2.2 GINAC_DECLARE_PRINT_CONTEXT_BASE . . . 1356
9.75.2.3 GINAC_DECLARE_PRINT_CONTEXT . . . 1356
9.75.2.4 GINAC_IMPLEMENT_PRINT_CONTEXT . . . 1357
9.76 pseries.cpp File Reference . . . 1357
9.76.1 Detailed Description . . . 1357
9.77 pseries.h File Reference . . . 1358
9.77.1 Detailed Description . . . 1358
9.78 ptr.h File Reference . . . 1358
9.78.1 Detailed Description . . . 1359
9.79 registrar.cpp File Reference . . . 1359
9.79.1 Detailed Description . . . 1359
9.80 registrar.h File Reference . . . 1359
9.80.1 Detailed Description . . . 1360
9.80.2 Macro Definition Documentation . . . 1360
9.80.2.1 GINAC_DECLARE_REGISTERED_CLASS_COMMON . . . 1360
9.80.2.2 GINAC_DECLARE_REGISTERED_CLASS_NO_CTORS . . . 1361
9.80.2.3 GINAC_DECLARE_REGISTERED_CLASS . . . 1361
9.80.2.4 GINAC_IMPLEMENT_REGISTERED_CLASS . . . 1361
9.80.2.5 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT . . . 1362
9.80.2.6 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT_T . . . 1362
9.81 relational.cpp File Reference . . . 1362
9.81.1 Detailed Description . . . 1363
9.82 relational.h File Reference . . . 1363
9.82.1 Detailed Description . . . 1363
9.83 remember.cpp File Reference . . . 1363
9.83.1 Detailed Description . . . 1364
9.84 remember.h File Reference . . . 1364
9.84.1 Detailed Description . . . 1364
9.85 structure.h File Reference . . . 1364
9.85.1 Detailed Description . . . 1365
9.86 symbol.cpp File Reference . . . 1365
9.86.1 Detailed Description . . . 1366
9.87 symbol.h File Reference . . . 1366

9.87.1 Detailed Description . . . 1367
9.88 symmetry.cpp File Reference . . . 1367
9.88.1 Detailed Description . . . 1368
9.89 symmetry.h File Reference . . . 1368
9.89.1 Detailed Description . . . 1369
9.90 tensor.cpp File Reference . . . 1369
9.90.1 Detailed Description . . . 1370
9.91 tensor.h File Reference . . . 1370
9.91.1 Detailed Description . . . 1371
9.92 utils.cpp File Reference . . . 1371
9.92.1 Detailed Description . . . 1374
9.93 utils.h File Reference . . . 1374
9.93.1 Detailed Description . . . 1375
9.93.2 Macro Definition Documentation . . . 1375
9.93.2.1 DEFAULT_CTOR . . . 1375
9.93.2.2 DEFAULT_COMPARE . . . 1375
9.93.2.3 DEFAULT_PRINT . . . 1376
9.93.2.4 DEFAULT_PRINT_LATEX . . . 1376
9.94 utils_multi_iterator.h File Reference . . . 1376
9.94.1 Detailed Description . . . 1378
9.95 version.h File Reference . . . 1378
9.95.1 Detailed Description . . . 1378
9.95.2 Macro Definition Documentation . . . 1378
9.95.2.1 GINACLIB_MAJOR_VERSION . . . 1378
9.95.2.2 GINACLIB_MINOR_VERSION . . . 1378
9.95.2.3 GINACLIB_MICRO_VERSION . . . 1378
9.95.2.4 GINAC_LT_CURRENT . . . 1379
9.95.2.5 GINAC_LT_REVISION . . . 1379
9.95.2.6 GINAC_LT_AGE . . . 1379
9.95.2.7 GINACLIB_ARCHIVE_VERSION . . . 1379
9.95.2.8 GINACLIB_ARCHIVE_AGE . . . 1379
9.95.2.9 GINACLIB_STR_HELPER . . . 1379
9.95.2.10 GINACLIB_STR . . . 1379
9.95.2.11 GINACLIB_VERSION . . . 1379
9.96 wildcard.cpp File Reference . . . 1380
9.96.1 Detailed Description . . . 1380
9.97 wildcard.h File Reference . . . 1380
9.97.1 Detailed Description . . . 1380

Chapter 1

Directory Hierarchy

1.1 Directories

ginac 23
add.cpp 1261
add.h 1262
archive.cpp 1262
archive.h 1263
assertion.h 1265
basic.cpp 1267
basic.h 1268
class_info.h 1269
clifford.cpp 1270
clifford.h 1272
color.cpp 1274
color.h 1276
compiler.h 1277
constant.cpp 1278
constant.h 1279
container.h 1280
crc32.h 1280
ex.cpp 1281
ex.h 1281
excompiler.cpp 1284
excompiler.h 1284
expair.cpp 1285
expair.h 1286
expairseq.cpp 1287
expairseq.h 1287
exprseq.cpp 1288
exprseq.h 1289
factor.cpp 1289
factor.h 1291
fail.cpp 1291
fail.h 1292
fderivative.cpp 1292
fderivative.h 1293
flags.h 1293
function.cpp 1294
function.h 1295
ginac.h 1306

Gt.cpp . . . 1307
Gt.h . . . 1308
Gt_helpers.cpp . . . 1308
Gt_helpers.h . . . 1309
hash_map.h . . . 1310
hash_seed.h . . . 1310
idx.cpp . . . 1311
idx.h . . . 1312
indexed.cpp . . . 1313
indexed.h . . . 1315
inifcns.cpp . . . 1316
inifcns.h . . . 1319
inifcns_elliptic.cpp . . . 1320
inifcns_gamma.cpp . . . 1322
inifcns_nstdsums.cpp . . . 1323
inifcns_trans.cpp . . . 1325
integral.cpp . . . 1328
integral.h . . . 1329
integration_kernel.cpp . . . 1329
integration_kernel.h . . . 1331
lst.cpp . . . 1332
lst.h . . . 1333
matrix.cpp . . . 1333
matrix.h . . . 1334
mul.cpp . . . 1336
mul.h . . . 1336
ncmul.cpp . . . 1337
ncmul.h . . . 1338
normal.cpp . . . 1339
normal.h . . . 1341
numeric.cpp . . . 1343
numeric.h . . . 1346
operators.cpp . . . 1349
operators.h . . . 1351
power.cpp . . . 1352
power.h . . . 1353
print.cpp . . . 1354
print.h . . . 1354
pseries.cpp . . . 1357
pseries.h . . . 1358
ptr.h . . . 1358
registrar.cpp . . . 1359
registrar.h . . . 1359
relational.cpp . . . 1362
relational.h . . . 1363
remember.cpp . . . 1363
remember.h . . . 1364
structure.h . . . 1364
symbol.cpp . . . 1365
symbol.h . . . 1366
symmetry.cpp . . . 1367
symmetry.h . . . 1368
tensor.cpp . . . 1369
tensor.h . . . 1370
utils.cpp . . . 1371
utils.h . . . 1374
utils_multi_iterator.h . . . 1376
version.h . . . 1378

wildcard.cpp 1380
wildcard.h 1380

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

GiNaC 29
GiNaC::Gt_detail 311
GiNaC::internal 313
std 313

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

GiNaC::internal::_iter_rep . . . 315
GiNaC::_numeric_digits . . . 317
GiNaC::archive . . . 337
GiNaC::archive_node . . . 344
GiNaC::archive_node::archive_node_cit_range . . . 355
GiNaC::archive::archived_ex . . . 357
GiNaC::basic_multi_iterator< T > . . . 394
  GiNaC::multi_iterator_counter< T > . . . 872
  GiNaC::multi_iterator_counter_indv< T > . . . 875
  GiNaC::multi_iterator_ordered< T > . . . 879
  GiNaC::multi_iterator_ordered_eq< T > . . . 883
  GiNaC::multi_iterator_ordered_eq_indv< T > . . . 887
  GiNaC::multi_iterator_permutation< T > . . . 891
  GiNaC::multi_iterator_shuffle< T > . . . 895
    GiNaC::multi_iterator_shuffle_prime< T > . . . 899
GiNaC::basic_partition_generator . . . 400
  GiNaC::partition_generator . . . 955
  GiNaC::partition_with_zero_parts_generator . . . 957
GiNaC::class_info< OPT > . . . 402
GiNaC::compare_all_equal< T > . . . 437
GiNaC::compare_bitwise< T > . . . 438
GiNaC::compare_std_less< T > . . . 439
ComparisonPolicy
  GiNaC::structure< T, ComparisonPolicy > . . . 1130
GiNaC::composition_generator . . . 440
GiNaC::const_iterator . . . 442
GiNaC::const_postorder_iterator . . . 449
GiNaC::const_preorder_iterator . . . 452
GiNaC::container_storage< C > . . . 481
  GiNaC::container< std::vector > . . . 466
    GiNaC::function . . . 644
      GiNaC::fderivative . . . 630
    GiNaC::indexed . . . 741
      GiNaC::clifford . . . 406

GiNaC::color . . . 425
GiNaC::ncmul . . . 911
GiNaC::container< std::list > . . . 466
GiNaC::container< C > . . . 466
GiNaC::composition_generator::coolmulti . . . 485
GiNaC::determinant_algo . . . 489
GiNaC::do_taylor . . . 516
GiNaC::domain . . . 516
std::domain_error
GiNaC::pole_error . . . 988
GiNaC::dunno . . . 517
GiNaC::composition_generator::coolmulti::element . . . 545
std::equal_to< GiNaC::ex > . . . 555
GiNaC::error_and_integral . . . 556
GiNaC::error_and_integral_is_less . . . 557
GiNaC::ex . . . 562
GiNaC::ex_base_is_less . . . 595
GiNaC::ex_is_equal . . . 595
GiNaC::ex_is_less . . . 596
GiNaC::ex_swap . . . 596
GiNaC::expair . . . 597
GiNaC::expair_is_less . . . 602
GiNaC::expair_rest_is_less . . . 602
GiNaC::expair_swap . . . 603
GiNaC::expand_options . . . 624
GiNaC::factor_options . . . 625
GiNaC::function_options . . . 665
GiNaC::G2_SERIAL . . . 706
GiNaC::G3_SERIAL . . . 706
GiNaC::gcd_options . . . 707
GiNaC::gcdheu_failed . . . 708
GiNaC::has_distance< T > . . . 724
GiNaC::has_options . . . 726
std::hash< GiNaC::ex > . . . 727
GiNaC::idx_is_equal_ignore_dim . . . 740
GiNaC::info_flags . . . 760
GiNaC::is_not_a_clifford . . . 782
GiNaC::is_summation_idx . . . 783
GiNaC::iterated_integral2_SERIAL . . . 783
GiNaC::iterated_integral3_SERIAL . . . 784
GiNaC::Gt::kernel . . . 784
GiNaC::lanczos_coeffs . . . 804
std::less< GiNaC::ptr< T > > . . . 806
GiNaC::library_init . . . 807
std::list
GiNaC::remember_table_list . . . 1101
GiNaC::make_flat_inserter . . . 809
GiNaC::map_function . . . 811
GiNaC::derivative_map_function . . . 487
GiNaC::eval_integ_map_function . . . 558
GiNaC::evalf_map_function . . . 559
GiNaC::evalm_map_function . . . 561
GiNaC::expand_map_function . . . 622
GiNaC::normal_map_function . . . 924
GiNaC::pointer_to_map_function . . . 966
GiNaC::pointer_to_map_function_1arg< T1 > . . . 968
GiNaC::pointer_to_map_function_2args< T1, T2 > . . . 971
GiNaC::pointer_to_map_function_3args< T1, T2, T3 > . . . 973

GiNaC::pointer_to_member_to_map_function< C > . . . 976
GiNaC::pointer_to_member_to_map_function_1arg< C, T1 > . . . 979
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 > . . . 982
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 > . . . 985
GiNaC::basic_partition_generator::mpartition2 . . . 852
GiNaC::op0_is_equal . . . 954
GiNaC::Gt_detail::pathintegral_term . . . 960
GiNaC::Gt_detail::pathintegral_term_list . . . 964
GiNaC::print_context . . . 1013
GiNaC::print_csrc . . . 1017
GiNaC::print_csrc_cl_N . . . 1019
GiNaC::print_csrc_double . . . 1021
GiNaC::print_csrc_float . . . 1023
GiNaC::print_dflt . . . 1025
GiNaC::print_latex . . . 1030
GiNaC::print_python . . . 1038
GiNaC::print_python_repr . . . 1039
GiNaC::print_tree . . . 1041
GiNaC::print_context_options . . . 1015
GiNaC::print_functor . . . 1026
GiNaC::print_functor_impl . . . 1029
GiNaC::print_memfun_handler< T, C > . . . 1032
GiNaC::print_ptrfun_handler< T, C > . . . 1035
GiNaC::print_options . . . 1034
GiNaC::archive_node::property . . . 1043
GiNaC::archive_node::property_info . . . 1044
GiNaC::psi1_SERIAL . . . 1063
GiNaC::psi2_SERIAL . . . 1064
GiNaC::ptr< T > . . . 1064
GiNaC::refcounted . . . 1076
GiNaC::basic . . . 358
GiNaC::container< std::vector > . . . 466
GiNaC::container< std::list > . . . 466
GiNaC::Gt . . . 708
GiNaC::constant . . . 455
GiNaC::container< C > . . . 466
GiNaC::expairseq . . . 604
GiNaC::add . . . 320
GiNaC::mul . . . 853
GiNaC::fail . . . 626
GiNaC::idx . . . 728
GiNaC::varidx . . . 1243
GiNaC::spinidx . . . 1110
GiNaC::integral . . . 761
GiNaC::integration_kernel . . . 773
GiNaC::ELi_kernel . . . 547
GiNaC::Ebar_kernel . . . 517
GiNaC::Eisenstein_h_kernel . . . 525
GiNaC::Eisenstein_kernel . . . 536
GiNaC::Kronecker_dtau_kernel . . . 787
GiNaC::Kronecker_dz_kernel . . . 796
GiNaC::basic_log_kernel . . . 388
GiNaC::modular_form_kernel . . . 843
GiNaC::multiple_polylog_kernel . . . 903
GiNaC::user_defined_kernel . . . 1235
GiNaC::matrix . . . 814
GiNaC::numeric . . . 925

GiNaC::power . . . 996
GiNaC::pseries . . . 1046
GiNaC::relational . . . 1082
GiNaC::structure< T, ComparisonPolicy > . . . 1130
GiNaC::symbol . . . 1175
GiNaC::realsymbol . . . 1069
GiNaC::possymbol . . . 990
GiNaC::symmetry . . . 1189
GiNaC::tensor . . . 1222
GiNaC::cliffordunit . . . 420
GiNaC::diracgamma . . . 490
GiNaC::diracgamma5 . . . 496
GiNaC::diracgammaL . . . 501
GiNaC::diracgammaR . . . 506
GiNaC::diracone . . . 511
GiNaC::su3d . . . 1148
GiNaC::su3f . . . 1153
GiNaC::su3one . . . 1159
GiNaC::su3t . . . 1164
GiNaC::tensdelta . . . 1202
GiNaC::tensepsilon . . . 1208
GiNaC::tensmetric . . . 1216
GiNaC::minkmetric . . . 836
GiNaC::spinmetric . . . 1120
GiNaC::wildcard . . . 1252
GiNaC::registered_class_options . . . 1079
GiNaC::remember_strategies . . . 1093
GiNaC::remember_table_entry . . . 1098
GiNaC::return_type_t . . . 1103
GiNaC::return_types . . . 1104
GiNaC::relational::safe_bool_helper . . . 1105
GiNaC::scalar_products . . . 1105
GiNaC::series_options . . . 1108
GiNaC::solve_algo . . . 1109
GiNaC::spmapkey . . . 1126
GiNaC::status_flags . . . 1129
GiNaC::subs_options . . . 1169
GiNaC::sy_is_less . . . 1170
GiNaC::sy_swap . . . 1171
GiNaC::sym_desc . . . 1172
GiNaC::symbolset . . . 1187
GiNaC::symminfo . . . 1199
GiNaC::symminfo_is_less_by_orig . . . 1201
GiNaC::symminfo_is_less_by_symmterm . . . 1202
GiNaC::terminfo . . . 1227
GiNaC::terminfo_is_less . . . 1229
GiNaC::Gt_detail::TransformExpressionWithCache< Type > . . . 1229
GiNaC::class_info< OPT >tree_node . . . 1232
GiNaC::unarchive_table_t . . . 1234
std::vector
GiNaC::remember_table . . . 1094
GiNaC::visitor . . . 1252
GiNaC::zeta1_SERIAL . . . 1259
GiNaC::zeta2_SERIAL . . . 1259

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

GiNaC::internal::_iter_rep . . . 315
GiNaC::_numeric_digits
This class is used to instantiate a global singleton object Digits which behaves just like Maple's Digits . . . 317
GiNaC::add
Sum of expressions . . . 320
GiNaC::archive
This class holds archived versions of GiNaC expressions (class ex) . . . 337
GiNaC::archive_node
This class stores all properties needed to record/retrieve the state of one object of class basic (or a derived class) . . . 344
GiNaC::archive_node::archive_node_cit_range . . . 355
GiNaC::archive::archived_ex
Archived expression descriptor . . . 357
GiNaC::basic
This class is the ABC (abstract base class) of GiNaC's class hierarchy . . . 358
GiNaC::basic_log_kernel
The basic integration kernel with a logarithmic singularity at the origin . . . 388
GiNaC::basic_multi_iterator< T >
Basic_multi_iterator is a base class . . . 394
GiNaC::basic_partition_generator
Base class for generating all bounded combinatorial partitions of an integer n with exactly m parts in non-decreasing order . . . 400
GiNaC::class_info< OPT > . . . 402
GiNaC::clifford
This class holds an object representing an element of the Clifford algebra (the Dirac gamma matrices) . . . 406
GiNaC::cliffordunit
This class represents the Clifford algebra generators (units) . . . 420
GiNaC::color
This class holds a generator T_a or the unity element of the Lie algebra of SU(3), as used for calculations in quantum chromodynamics . . . 425
GiNaC::compare_all_equal< T >
Comparison policy: all structures of one type are equal . . . 437
GiNaC::compare_bitwise< T >
Comparison policy: use bit-wise comparison to compare structures . . . 438

GiNaC::compare_std_less< T >
Comparison policy: use std::equal_to/std::less (defaults to operators == and <) to compare structures . . . 439
GiNaC::composition_generator
Generate all compositions of a partition of an integer n, starting with the compositions which has non-decreasing order . . . 440
GiNaC::const_iterator . . . 442
GiNaC::const_postorder_iterator . . . 449
GiNaC::const_preorder_iterator . . . 452
GiNaC::constant
This class holds constants, symbols with specific numerical value . . . 455
GiNaC::container< C >
Wrapper template for making GiNaC classes out of STL containers . . . 466
GiNaC::container_storage< C >
Helper template for encapsulating the reserve() mechanics of STL containers . . . 481
GiNaC::composition_generator::coolmulti . . . 485
GiNaC::derivative_map_function
Function object to be applied by basic::derivative() . . . 487
GiNaC::determinant_algo
Switch to control algorithm for determinant computation . . . 489
GiNaC::diracgamma
This class represents the Dirac gamma Lorentz vector . . . 490
GiNaC::diracgamma5
This class represents the Dirac gamma5 object which anticommutates with all other gammas . . . 496
GiNaC::diracgammaL
This class represents the Dirac gammaL object which behaves like 1/2 (1-gamma5) . . . 501
GiNaC::diracgammaR
This class represents the Dirac gammaL object which behaves like 1/2 (1+gamma5) . . . 506
GiNaC::diracone
This class represents the Clifford algebra unity element . . . 511
GiNaC::do_taylor
Exception class thrown by classes which provide their own series expansion to signal that ordinary Taylor expansion is safe . . . 516
GiNaC::domain
Domain of an object . . . 516
GiNaC::dunno
Exception class thrown by functions to signal unimplemented functionality so the expression may just be .hold() . . . 517
GiNaC::Ebar_kernel
The Ebar-kernel . . . 517
GiNaC::Eisenstein_h_kernel
The kernel corresponding to the Eisenstein series $h_{k,N,r,s}(\tau)$ . . . 525
GiNaC::Eisenstein_kernel
The kernel corresponding to the Eisenstein series $E_{k,N,a,b,K}(\tau)$ . . . 536
GiNaC::composition_generator::coolmulti::element . . . 545
GiNaC::ELi_kernel
The ELi-kernel . . . 547
std::equal_to< GiNaC::ex >
Specialization of std::equal_to() for ex objects . . . 555
GiNaC::error_and_integral . . . 556
GiNaC::error_and_integral_is_less . . . 557
GiNaC::eval_integ_map_function
Function object to be applied by basic::eval_integ() . . . 558
GiNaC::evalf_map_function
Function object to be applied by basic::evalf() . . . 559
GiNaC::evalm_map_function
Function object to be applied by basic::evalm() . . . 561

GiNaC::ex
Lightweight wrapper for GiNaC's symbolic objects . . . 562
GiNaC::ex_base_is_less . . . 595
GiNaC::ex_is_equal . . . 595
GiNaC::ex_is_less . . . 596
GiNaC::ex_swap . . . 596
GiNaC::expair
A pair of expressions . . . 597
GiNaC::expair_is_less
Function object for insertion into third argument of STL's sort() etc . . . 602
GiNaC::expair_rest_is_less
Function object not caring about the numerical coefficients for insertion into third argument of STL's sort() . . . 602
GiNaC::expair_swap . . . 603
GiNaC::expairseq
A sequence of class expair . . . 604
GiNaC::expand_map_function
Function object to be applied by basic::expand() . . . 622
GiNaC::expand_options
Flags to control the behavior of expand() . . . 624
GiNaC::factor_options
Flags to control the polynomial factorization . . . 625
GiNaC::fail . . . 626
GiNaC::fderivative
This class represents the (abstract) derivative of a symbolic function . . . 630
GiNaC::function
The class function is used to implement builtin functions like sin, cos... and user defined functions 644
GiNaC::function_options . . . 665
GiNaC::G2_SERIAL
Generalized multiple polylogarithm . . . 706
GiNaC::G3_SERIAL
Generalized multiple polylogarithm with explicit imaginary parts . . . 706
GiNaC::gcd_options
Flags to control the behavior of gcd() and friends . . . 707
GiNaC::gcdheu_failed
Exception thrown by heur_gcd() to signal failure . . . 708
GiNaC::Gt
Elliptic Multiple Polylogarithm . . . 708
GiNaC::has_distance< T >
SFINAE test for distance . . . 724
GiNaC::has_options
Flags to control the behavior of has() . . . 726
std::hash< GiNaC::ex >
Specialization of std::hash() for ex objects . . . 727
GiNaC::idx
This class holds one index of an indexed object . . . 728
GiNaC::idx_is_equal_ignore_dim . . . 740
GiNaC::indexed
This class holds an indexed expression . . . 741
GiNaC::info_flags
Possible attributes an object can have . . . 760
GiNaC::integral
Symbolic integral . . . 761
GiNaC::integration_kernel
The base class for integration kernels for iterated integrals . . . 773
GiNaC::is_not_a_clifford
Predicate for finding non-clifford objects . . . 782
GiNaC::is_summation_idx . . . 783

GiNaC::iterated_integral2_SERIAL
Complete elliptic integral of the first kind . . . 783
GiNaC::iterated_integral3_SERIAL
Iterated integral with explicit truncation . . . 784
GiNaC::Gt::kernel . . . 784
GiNaC::Kronecker_dtau_kernel
The kernel corresponding to integrating the Kronecker coefficient function $g^{(n)}(z_j, K\tau)$ in τ (or equivalently in $\bar{q}$) . . . 787
GiNaC::Kronecker_dz_kernel
The kernel corresponding to integrating the Kronecker coefficient function $g^{(n-1)}(z - z_j, K\tau)$ in z . . . 796
GiNaC::lanczos_coeffs . . . 804
std::less< GiNaC::ptr< T > >
Specialization of std::less for ptr<T> to enable ordering of ptr<T> objects (e.g . . . 806
GiNaC::library_init
Helper class to initialize the library . . . 807
GiNaC::make_flat_inserter
Class to handle the renaming of dummy indices . . . 809
GiNaC::map_function
Function object for map() . . . 811
GiNaC::matrix
Symbolic matrices . . . 814
GiNaC::minkmetric
This class represents a Minkowski metric tensor . . . 836
GiNaC::modular_form_kernel
A kernel corresponding to a polynomial in Eisenstein series . . . 843
GiNaC::basic_partition_generator::mpartition2 . . . 852
GiNaC::mul
Product of expressions . . . 853
GiNaC::multi_iterator_counter< T >
The class multi_iterator_counter defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that . . . 872
GiNaC::multi_iterator_counter_indv< T >
The class multi_iterator_counter_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that . . . 875
GiNaC::multi_iterator_ordered< T >
The class multi_iterator_ordered defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that . . . 879
GiNaC::multi_iterator_ordered_eq< T >
The class multi_iterator_ordered_eq defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that . . . 883
GiNaC::multi_iterator_ordered_eq_indv< T >
The class multi_iterator_ordered_eq_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that . . . 887
GiNaC::multi_iterator_permutation< T >
The class multi_iterator_permutation defines a multi_iterator $(i_1, i_2, ..., i_k)$, for which . . . 891
GiNaC::multi_iterator_shuffle< T >
The class multi_iterator_shuffle defines a multi_iterator, which runs over all shuffles of a and b 895
GiNaC::multi_iterator_shuffle_prime< T >
The class multi_iterator_shuffle_prime defines a multi_iterator, which runs over all shuffles of a and b, excluding the first one (a,b) . . . 899
GiNaC::multiple_polylog_kernel
The integration kernel for multiple polylogarithms . . . 903
GiNaC::ncmul
Non-commutative product of expressions . . . 911
GiNaC::normal_map_function
Function object to be applied by basic::normal() . . . 924
GiNaC::numeric
This class is a wrapper around CLN-numbers within the GiNaC class hierarchy . . . 925
GiNaC::op0_is_equal . . . 954
GiNaC::partition_generator
Generate all bounded combinatorial partitions of an integer n with exactly m parts (not including zero parts) in non-decreasing order . . . 955

GiNaC::partition_with_zero_parts_generator
Generate all bounded combinatorial partitions of an integer n with exactly m parts (including zero parts) in non-decreasing order . . . 957
GiNaC::Gt_detail::pathintegral_term . . . 960
GiNaC::Gt_detail::pathintegral_term_list . . . 964
GiNaC::pointer_to_map_function . . . 966
GiNaC::pointer_to_map_function_1arg< T1 > . . . 968
GiNaC::pointer_to_map_function_2args< T1, T2 > . . . 971
GiNaC::pointer_to_map_function_3args< T1, T2, T3 > . . . 973
GiNaC::pointer_to_member_to_map_function< C > . . . 976
GiNaC::pointer_to_member_to_map_function_1arg< C, T1 > . . . 979
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 > . . . 982
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 > . . . 985
GiNaC::pole_error
Exception class thrown when a singularity is encountered . . . 988
GiNaC::possymbol
Specialization of symbol to real positive domain . . . 990
GiNaC::power
This class holds a two-component object, a basis and and exponent representing exponentiation 996
GiNaC::print_context
Base class for print_contexts . . . 1013
GiNaC::print_context_options
This class stores information about a registered print_context class . . . 1015
GiNaC::print_csrc
Base context for C source output . . . 1017
GiNaC::print_csrc_cl_N
Context for C source output using CLN numbers . . . 1019
GiNaC::print_csrc_double
Context for C source output using double precision . . . 1021
GiNaC::print_csrc_float
Context for C source output using float precision . . . 1023
GiNaC::print_dflt
Context for default (ginsh-parsable) output . . . 1025
GiNaC::print_functor
This class represents a print method for a certain algebraic class and print_context type . . . 1026
GiNaC::print_functor_impl
Base class for print_functor handlers . . . 1029
GiNaC::print_latex
Context for latex-parsable output . . . 1030
GiNaC::print_memfun_handler< T, C >
Print_functor handler for member functions of class T, context type C . . . 1032
GiNaC::print_options
Flags to control the behavior of a print_context . . . 1034
GiNaC::print_ptrfun_handler< T, C >
Print_functor handler for pointer-to-functions of class T, context type C . . . 1035
GiNaC::print_python
Context for python pretty-print output . . . 1038
GiNaC::print_python_repr
Context for python-parsable output . . . 1039
GiNaC::print_tree
Context for tree-like output for debugging . . . 1041
GiNaC::archive_node::property
Archived property (data type, name and associated data) . . . 1043
GiNaC::archive_node::property_info
Information about a stored property . . . 1044
GiNaC::pseries
This class holds a extended truncated power series (positive and negative integer powers) . . 1046

GiNaC::psi1_SERIAL
Polylogarithm and multiple polylogarithm . . . 1063
GiNaC::psi2_SERIAL
Derivatives of Psi-function (aka polygamma-functions) . . . 1064
GiNaC::ptr< T >
Class of (intrusively) reference-counted pointers that support copy-on-write semantics . . . 1064
GiNaC::realsymbol
Specialization of symbol to real domain . . . 1069
GiNaC::refcounted
Base class for reference-counted objects . . . 1076
GiNaC::registered_class_options
This class stores information about a registered GiNaC class . . . 1079
GiNaC::relational
This class holds a relation consisting of two expressions and a logical relation between them . . . 1082
GiNaC::remember_strategies
Strategies how to clean up the function remember cache . . . 1093
GiNaC::remember_table
The remember table is organized like an n-fold associative cache in a microprocessor . . . 1094
GiNaC::remember_table_entry
A single entry in the remember table of a function . . . 1098
GiNaC::remember_table_list
A list of entries in the remember table having some least significant bits of the hashvalue in common . . . 1101
GiNaC::return_type_t
To distinguish between different kinds of non-commutative objects . . . 1103
GiNaC::return_types . . . 1104
GiNaC::relational::safe_bool_helper . . . 1105
GiNaC::scalar_products
Helper class for storing information about known scalar products which are to be automatically replaced by simplify_indexed() . . . 1105
GiNaC::series_options
Flags to control series expansion . . . 1108
GiNaC::solve_algo
Switch to control algorithm for linear system solving . . . 1109
GiNaC::spinidx
This class holds a spinor index that can be dotted or undotted and that also has a variance . . . 1110
GiNaC::spinmetric
This class represents an antisymmetric spinor metric tensor which can be used to raise/lower indices of 2-component Weyl spinors . . . 1120
GiNaC::spmapkey . . . 1126
GiNaC::status_flags
Flags to store information about the state of an object . . . 1129
GiNaC::structure< T, ComparisonPolicy >
Wrapper template for making GiNaC classes out of C++ structures . . . 1130
GiNaC::su3d
This class represents the tensor of symmetric su(3) structure constants . . . 1148
GiNaC::su3f
This class represents the tensor of antisymmetric su(3) structure constants . . . 1153
GiNaC::su3one
This class represents the su(3) unity element . . . 1159
GiNaC::su3t
This class represents an su(3) generator . . . 1164
GiNaC::subs_options
Flags to control the behavior of subs() . . . 1169
GiNaC::sy_is_less . . . 1170
GiNaC::sy_swap . . . 1171

GiNaC::sym_desc
This structure holds information about the highest and lowest degrees in which a symbol appears in two multivariate polynomials "a" and "b" . . . 1172
GiNaC::symbol
Basic CAS symbol . . . 1175
GiNaC::symbolset . . . 1187
GiNaC::symmetry
This class describes the symmetry of a group of indices . . . 1189
GiNaC::symminfo
This structure stores the individual symmetrized terms obtained during the simplification of sums 1199
GiNaC::symminfo_is_less_by_orig . . . 1201
GiNaC::symminfo_is_less_by_symmterm . . . 1202
GiNaC::tensdelta
This class represents the delta tensor . . . 1202
GiNaC::tensepsilon
This class represents the totally antisymmetric epsilon tensor . . . 1208
GiNaC::tensmetric
This class represents a general metric tensor which can be used to raise/lower indices . . . 1216
GiNaC::tensor
This class holds one of GiNaC's predefined special tensors such as the delta and the metric tensors . . . 1222
GiNaC::terminfo
This structure stores the original and symmetrized versions of terms obtained during the simplification of sums . . . 1227
GiNaC::terminfo_is_less . . . 1229
GiNaC::Gt_detail::TransformExpressionWithCache< Type > . . . 1229
GiNaC::class_info< OPT >::tree_node . . . 1232
GiNaC::unarchive_table_t . . . 1234
GiNaC::user_defined_kernel
A user-defined integration kernel . . . 1235
GiNaC::varidx
This class holds an index with a variance (co- or contravariant) . . . 1243
GiNaC::visitor
Degenerate base class for visitors . . . 1252
GiNaC::wildcard
This class acts as a wildcard for subs(), match(), has() and find() . . . 1252
GiNaC::zeta1_SERIAL
Complex conjugate . . . 1259
GiNaC::zeta2_SERIAL
Alternating Euler sum or colored MZV . . . 1259

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

add.cpp
Implementation of GiNaC's sums of expressions . . . 1261
add.h
Interface to GiNaC's sums of expressions . . . 1262
archive.cpp
Archiving of GiNaC expressions . . . 1262
archive.h
Archiving of GiNaC expressions . . . 1263
assertion.h
Assertion macro definition . . . 1265
basic.cpp
Implementation of GiNaC's ABC . . . 1267
basic.h
Interface to GiNaC's ABC . . . 1268
class_info.h
Helper templates to provide per-class information for class hierarchies . . . 1269
clifford.cpp
Implementation of GiNaC's clifford algebra (Dirac gamma) objects . . . 1270
clifford.h
Interface to GiNaC's clifford algebra (Dirac gamma) objects . . . 1272
color.cpp
Implementation of GiNaC's color (SU(3) Lie algebra) objects . . . 1274
color.h
Interface to GiNaC's color (SU(3) Lie algebra) objects . . . 1276
compiler.h
Definition of optimizing macros . . . 1277
constant.cpp
Implementation of GiNaC's constant types and some special constants . . . 1278
constant.h
Interface to GiNaC's constant types and some special constants . . . 1279
container.h
Wrapper template for making GiNaC classes out of STL containers . . . 1280
crc32.h
CRC32 hash function . . . 1280
ex.cpp
Implementation of GiNaC's light-weight expression handles . . . 1281

ex.h
Interface to GiNaC's light-weight expression handles . . . 1281
excompiler.cpp
Functions to facilitate the conversion of a ex to a function pointer suited for fast numerical integration . . . 1284
excompiler.h
Functions to facilitate the conversion of a ex to a function pointer suited for fast numerical integration . . . 1284
expair.cpp
Implementation of expression pairs (building blocks of expairseq) . . . 1285
expair.h
Definition of expression pairs (building blocks of expairseq) . . . 1286
expairseq.cpp
Implementation of sequences of expression pairs . . . 1287
expairseq.h
Interface to sequences of expression pairs . . . 1287
exprseq.cpp
Implementation of GiNaC's exprseq . . . 1288
exprseq.h
Definition of GiNaC's exprseq . . . 1289
factor.cpp
Polynomial factorization (implementation) . . . 1289
factor.h
Polynomial factorization . . . 1291
fail.cpp
Implementation of class signaling failure of operation . . . 1291
fail.h
Interface to class signaling failure of operation . . . 1292
fderivative.cpp
Implementation of abstract derivatives of functions . . . 1292
fderivative.h
Interface to abstract derivatives of functions . . . 1293
flags.h
Collection of all flags used through the GiNaC framework . . . 1293
function.cpp
Implementation of class of symbolic functions . . . 1294
function.h
Interface to class of symbolic functions . . . 1295
ginac.h
This include file includes all other public GiNaC headers . . . 1306
Gt.cpp
Implementation of GiNaC's elliptic multiple polylogarithms . . . 1307
Gt.h
Interface to GiNaC's elliptic multiple polylogarithms . . . 1308
Gt_helpers.cpp
Implementation of helper functions for elliptic multiple polylogarithms . . . 1308
Gt_helpers.h
Interface to helper functions for elliptic multiple polylogarithms . . . 1309
hash_map.h
Replacement for map<> using hash tables . . . 1310
hash_seed.h
Type-specific hash seed . . . 1310
idx.cpp
Implementation of GiNaC's indices . . . 1311
idx.h
Interface to GiNaC's indices . . . 1312
indexed.cpp
Implementation of GiNaC's indexed expressions . . . 1313

indexed.h
Interface to GiNaC's indexed expressions . . . 1315
inifcns.cpp
Implementation of GiNaC's initially known functions . . . 1316
inifcns.h
Interface to GiNaC's initially known functions . . . 1319
inifcns_elliptic.cpp
Implementation of some special functions related to elliptic curves . . . 1320
inifcns_gamma.cpp
Implementation of Gamma-function, Beta-function, Polygamma-functions, and some related stuff 1322
inifcns_nstdsums.cpp
Implementation of some special functions that have a representation as nested sums . . . 1323
inifcns_trans.cpp
Implementation of transcendental (and trigonometric and hyperbolic) functions . . . 1325
integral.cpp
Implementation of GiNaC's symbolic integral . . . 1328
integral.h
Interface to GiNaC's symbolic integral . . . 1329
integration_kernel.cpp
Implementation of GiNaC's integration kernels for iterated integrals . . . 1329
integration_kernel.h
Interface to GiNaC's integration kernels for iterated integrals . . . 1331
lst.cpp
Implementation of GiNaC's lst . . . 1332
lst.h
Definition of GiNaC's lst . . . 1333
matrix.cpp
Implementation of symbolic matrices . . . 1333
matrix.h
Interface to symbolic matrices . . . 1334
mul.cpp
Implementation of GiNaC's products of expressions . . . 1336
mul.h
Interface to GiNaC's products of expressions . . . 1336
ncmul.cpp
Implementation of GiNaC's non-commutative products of expressions . . . 1337
ncmul.h
Interface to GiNaC's non-commutative products of expressions . . . 1338
normal.cpp
This file implements several functions that work on univariate and multivariate polynomials and rational functions . . . 1339
normal.h
This file defines several functions that work on univariate and multivariate polynomials and rational functions . . . 1341
numeric.cpp
This file contains the interface to the underlying bignum package . . . 1343
numeric.h
Makes the interface to the underlying bignum package available . . . 1346
operators.cpp
Implementation of GiNaC's overloaded operators . . . 1349
operators.h
Interface to GiNaC's overloaded operators . . . 1351
power.cpp
Implementation of GiNaC's symbolic exponentiation (basis^exponent) . . . 1352
power.h
Interface to GiNaC's symbolic exponentiation (basis^exponent) . . . 1353
print.cpp
Implementation of helper classes for expression output . . . 1354

print.h
Definition of helper classes for expression output . . . 1354
pseries.cpp
Implementation of class for extended truncated power series and methods for series expansion 1357
pseries.h
Interface to class for extended truncated power series . . . 1358
ptr.h
Reference-counted pointer template . . . 1358
registrar.cpp
GiNaC's class registrar (for class basic and all classes derived from it) . . . 1359
registrar.h
GiNaC's class registrar (for class basic and all classes derived from it) . . . 1359
relational.cpp
Implementation of relations between expressions . . . 1362
relational.h
Interface to relations between expressions . . . 1363
remember.cpp
Implementation of helper classes for using the remember option in GiNaC functions . . . 1363
remember.h
Interface to helper classes for using the remember option in GiNaC functions . . . 1364
structure.h
Wrapper template for making GiNaC classes out of C++ structures . . . 1364
symbol.cpp
Implementation of GiNaC's symbolic objects . . . 1365
symbol.h
Interface to GiNaC's symbolic objects . . . 1366
symmetry.cpp
Implementation of GiNaC's symmetry definitions . . . 1367
symmetry.h
Interface to GiNaC's symmetry definitions . . . 1368
tensor.cpp
Implementation of GiNaC's special tensors . . . 1369
tensor.h
Interface to GiNaC's special tensors . . . 1370
utils.cpp
Implementation of several small and furry utilities needed within GiNaC but not of any interest to the user of the library . . . 1371
utils.h
Interface to several small and furry utilities needed within GiNaC but not of any interest to the user of the library . . . 1374
utils_multi_iterator.h
Utilities for summing over multiple indices . . . 1376
version.h
GiNaC library version information . . . 1378
wildcard.cpp
Implementation of GiNaC's wildcard objects . . . 1380
wildcard.h
Interface to GiNaC's wildcard objects . . . 1380

Chapter 6

Directory Documentation

6.1 ginac Directory Reference

Files

- file add.cpp

  Implementation of GiNaC's sums of expressions.
- file add.h

  Interface to GiNaC's sums of expressions.
- file archive.cpp

  Archiving of GiNaC expressions.
- file archive.h

  Archiving of GiNaC expressions.
- file assertion.h

  Assertion macro definition.
- file basic.cpp

  Implementation of GiNaC's ABC.
- file basic.h

  Interface to GiNaC's ABC.
- file class_info.h

  Helper templates to provide per-class information for class hierarchies.
- file clifford.cpp

  Implementation of GiNaC's clifford algebra (Dirac gamma) objects.
- file clifford.h

  Interface to GiNaC's clifford algebra (Dirac gamma) objects.
- file color.cpp

  Implementation of GiNaC's color (SU(3) Lie algebra) objects.
- file color.h

  Interface to GiNaC's color (SU(3) Lie algebra) objects.
- file compiler.h

  Definition of optimizing macros.
- file constant.cpp

  Implementation of GiNaC's constant types and some special constants.
- file constant.h

  Interface to GiNaC's constant types and some special constants.
- file container.h

Wrapper template for making GiNaC classes out of STL containers.

- file crc32.h

  CRC32 hash function.

- file ex.cpp

  Implementation of GiNaC's light-weight expression handles.

- file ex.h

  Interface to GiNaC's light-weight expression handles.

- file excompiler.cpp

  Functions to facilitate the conversion of a ex to a function pointer suited for fast numerical integration.

- file excompiler.h

  Functions to facilitate the conversion of a ex to a function pointer suited for fast numerical integration.

- file expair.cpp

  Implementation of expression pairs (building blocks of expairseq).

- file expair.h

  Definition of expression pairs (building blocks of expairseq).

- file expairseq.cpp

  Implementation of sequences of expression pairs.

- file expairseq.h

  Interface to sequences of expression pairs.

- file exprseq.cpp

  Implementation of GiNaC's exprseq.

- file exprseq.h

  Definition of GiNaC's exprseq.

- file factor.cpp

  Polynomial factorization (implementation).

- file factor.h

  Polynomial factorization.

- file fail.cpp

  Implementation of class signaling failure of operation.

- file fail.h

  Interface to class signaling failure of operation.

- file fderivative.cpp

  Implementation of abstract derivatives of functions.

- file fderivative.h

  Interface to abstract derivatives of functions.

- file flags.h

  Collection of all flags used through the GiNaC framework.

- file function.cpp

  Implementation of class of symbolic functions.

- file function.h

  Interface to class of symbolic functions.

- file ginac.h

  This include file includes all other public GiNaC headers.

- file Gt.cpp

  Implementation of GiNaC's elliptic multiple polylogarithms.

- file Gt.h

  Interface to GiNaC's elliptic multiple polylogarithms.

- file Gt_helpers.cpp

  Implementation of helper functions for elliptic multiple polylogarithms.

- file Gt_helpers.h

  Interface to helper functions for elliptic multiple polylogarithms.

- file hash_map.h

  Replacement for map<> using hash tables.
- file hash_seed.h

  Type-specific hash seed.
- file idx.cpp

  Implementation of GiNaC's indices.
- file idx.h

  Interface to GiNaC's indices.
- file indexed.cpp

  Implementation of GiNaC's indexed expressions.
- file indexed.h

  Interface to GiNaC's indexed expressions.
- file inifcns.cpp

  Implementation of GiNaC's initially known functions.
- file inifcns.h

  Interface to GiNaC's initially known functions.
- file inifcns_elliptic.cpp

  Implementation of some special functions related to elliptic curves.
- file inifcns_gamma.cpp

  Implementation of Gamma-function, Beta-function, Polygamma-functions, and some related stuff.
- file inifcns_nstdsums.cpp

  Implementation of some special functions that have a representation as nested sums.
- file inifcns_trans.cpp

  Implementation of transcendental (and trigonometric and hyperbolic) functions.
- file integral.cpp

  Implementation of GiNaC's symbolic integral.
- file integral.h

  Interface to GiNaC's symbolic integral.
- file integration_kernel.cpp

  Implementation of GiNaC's integration kernels for iterated integrals.
- file integration_kernel.h

  Interface to GiNaC's integration kernels for iterated integrals.
- file lst.cpp

  Implementation of GiNaC's lst.
- file lst.h

  Definition of GiNaC's lst.
- file matrix.cpp

  Implementation of symbolic matrices.
- file matrix.h

  Interface to symbolic matrices.
- file mul.cpp

  Implementation of GiNaC's products of expressions.
- file mul.h

  Interface to GiNaC's products of expressions.
- file ncmul.cpp

  Implementation of GiNaC's non-commutative products of expressions.
- file ncmul.h

  Interface to GiNaC's non-commutative products of expressions.
- file normal.cpp

  This file implements several functions that work on univariate and multivariate polynomials and rational functions.
- file normal.h

This file defines several functions that work on univariate and multivariate polynomials and rational functions.

- file numeric.cpp

  This file contains the interface to the underlying bignum package.
- file numeric.h

  Makes the interface to the underlying bignum package available.
- file operators.cpp

  Implementation of GiNaC's overloaded operators.
- file operators.h

  Interface to GiNaC's overloaded operators.
- file power.cpp

  Implementation of GiNaC's symbolic exponentiation (basis^ exponent).
- file power.h

  Interface to GiNaC's symbolic exponentiation (basis^ exponent).
- file print.cpp

  Implementation of helper classes for expression output.
- file print.h

  Definition of helper classes for expression output.
- file pseries.cpp

  Implementation of class for extended truncated power series and methods for series expansion.
- file pseries.h

  Interface to class for extended truncated power series.
- file ptr.h

  Reference-counted pointer template.
- file registrar.cpp

  GiNaC's class registrar (for class basic and all classes derived from it).
- file registrar.h

  GiNaC's class registrar (for class basic and all classes derived from it).
- file relational.cpp

  Implementation of relations between expressions.
- file relational.h

  Interface to relations between expressions.
- file remember.cpp

  Implementation of helper classes for using the remember option in GiNaC functions.
- file remember.h

  Interface to helper classes for using the remember option in GiNaC functions.
- file structure.h

  Wrapper template for making GiNaC classes out of C++ structures.
- file symbol.cpp

  Implementation of GiNaC's symbolic objects.
- file symbol.h

  Interface to GiNaC's symbolic objects.
- file symmetry.cpp

  Implementation of GiNaC's symmetry definitions.
- file symmetry.h

  Interface to GiNaC's symmetry definitions.
- file tensor.cpp

  Implementation of GiNaC's special tensors.
- file tensor.h

  Interface to GiNaC's special tensors.
- file utils.cpp

  Implementation of several small and furry utilities needed within GiNaC but not of any interest to the user of the library.

- file utils.h

  Interface to several small and furry utilities needed within GiNaC but not of any interest to the user of the library.
- file utils_multi_iterator.h

  Utilities for summing over multiple indices.
- file version.h

  GiNaC library version information.
- file wildcard.cpp

  Implementation of GiNaC's wildcard objects.
- file wildcard.h

  Interface to GiNaC's wildcard objects.

Chapter 7

Namespace Documentation

7.1 GiNaC Namespace Reference

Namespaces

- namespace Gt_detail
- namespace internal

Classes

- class _numeric_digits

  This class is used to instantiate a global singleton object Digits which behaves just like Maple's Digits.
- class add

  Sum of expressions.
- class archive

  This class holds archived versions of GiNaC expressions (class ex).
- class archive_node

  This class stores all properties needed to record/retrieve the state of one object of class basic (or a derived class).
- class basic

  This class is the ABC (abstract base class) of GiNaC's class hierarchy.
- class basic_log_kernel

  The basic integration kernel with a logarithmic singularity at the origin.
- class basic_multi_iterator

  basic_multi_iterator is a base class.
- class basic_partition_generator

  Base class for generating all bounded combinatorial partitions of an integer n with exactly m parts in non-decreasing order.
- class class_info
- class clifford

  This class holds an object representing an element of the Clifford algebra (the Dirac gamma matrices).
- class cliffordunit

  This class represents the Clifford algebra generators (units).
- class color

  This class holds a generator T_a or the unity element of the Lie algebra of SU(3), as used for calculations in quantum chromodynamics.
- class compare_all_equal

Comparison policy: all structures of one type are equal.

- class compare_bitwise

  Comparison policy: use bit-wise comparison to compare structures.
- class compare_std_less

  Comparison policy: use std::equal_to/std::less (defaults to operators == and <) to compare structures.
- class composition_generator

  Generate all compositions of a partition of an integer n, starting with the compositions which has non-decreasing order.
- class const_iterator
- class const_postorder_iterator
- class const_preorder_iterator
- class constant

  This class holds constants, symbols with specific numerical value.
- class container

  Wrapper template for making GiNaC classes out of STL containers.
- class container_storage

  Helper template for encapsulating the reserve() mechanics of STL containers.
- struct derivative_map_function

  Function object to be applied by basic::derivative().
- class determinant_algo

  Switch to control algorithm for determinant computation.
- class diracgamma

  This class represents the Dirac gamma Lorentz vector.
- class diracgamma5

  This class represents the Dirac gamma5 object which anticommutates with all other gammas.
- class diracgammaL

  This class represents the Dirac gammaL object which behaves like 1/2 (1-gamma5).
- class diracgammaR

  This class represents the Dirac gammaL object which behaves like 1/2 (1+gamma5).
- class diracone

  This class represents the Clifford algebra unity element.
- class do_taylor

  Exception class thrown by classes which provide their own series expansion to signal that ordinary Taylor expansion is safe.
- class domain

  Domain of an object.
- class dunno

  Exception class thrown by functions to signal unimplemented functionality so the expression may just be .hold().
- class Ebar_kernel

  The Ebar-kernel.
- class Eisenstein_h_kernel

  The kernel corresponding to the Eisenstein series $h_{k,N,r,s}(\tau)$.
- class Eisenstein_kernel

  The kernel corresponding to the Eisenstein series $E_{k,N,a,b,K}(\tau)$.
- class ELi_kernel

  The ELi-kernel.
- struct error_and_integral
- struct error_and_integral_is_less
- struct eval_integ_map_function

  Function object to be applied by basic::eval_integ().
- struct evalf_map_function

Function object to be applied by basic::evalf().

- struct evalm_map_function

  Function object to be applied by basic::evalm().
- class ex

  Lightweight wrapper for GiNaC's symbolic objects.
- struct ex_base_is_less
- struct ex_is_equal
- struct ex_is_less
- struct ex_swap
- class expair

  A pair of expressions.
- struct expair_is_less

  Function object for insertion into third argument of STL's sort() etc.
- struct expair_rest_is_less

  Function object not caring about the numerical coefficients for insertion into third argument of STL's sort().
- struct expair_swap
- class expairseq

  A sequence of class expair.
- struct expand_map_function

  Function object to be applied by basic::expand().
- class expand_options

  Flags to control the behavior of expand().
- class factor_options

  Flags to control the polynomial factorization.
- class fail
- class fderivative

  This class represents the (abstract) derivative of a symbolic function.
- class function

  The class function is used to implement builtin functions like sin, cos... and user defined functions.
- class function_options
- class G2_SERIAL

  Generalized multiple polylogarithm.
- class G3_SERIAL

  Generalized multiple polylogarithm with explicit imaginary parts.
- struct gcd_options

  Flags to control the behavior of gcd() and friends.
- class gcdheu_failed

  Exception thrown by heur_gcd() to signal failure.
- class Gt

  Elliptic Multiple Polylogarithm.
- class has_distance

  SFINAE test for distance.
- class has_options

  Flags to control the behavior of has().
- class idx

  This class holds one index of an indexed object.
- struct idx_is_equal_ignore_dim
- class indexed

  This class holds an indexed expression.
- class info_flags

  Possible attributes an object can have.

- class integral

  Symbolic integral.
- class integration_kernel

  The base class for integration kernels for iterated integrals.
- struct is_not_a_clifford

  Predicate for finding non-clifford objects.
- struct is_summation_idx
- class iterated_integral2_SERIAL

  Complete elliptic integral of the first kind.
- class iterated_integral3_SERIAL

  Iterated integral with explicit truncation.
- class Kronecker_dtau_kernel

  The kernel corresponding to integrating the Kronecker coefficient function $g^{(n)}(z_j, K\tau)$ in τ (or equivalently in $\bar{q}$).
- class Kronecker_dz_kernel

  The kernel corresponding to integrating the Kronecker coefficient function $g^{(n-1)}(z - z_j, K\tau)$ in z.
- class lanczos_coeffs
- class library_init

  Helper class to initialize the library.
- class make_flat_inserter

  Class to handle the renaming of dummy indices.
- struct map_function

  Function object for map().
- class matrix

  Symbolic matrices.
- class minkmetric

  This class represents a Minkowski metric tensor.
- class modular_form_kernel

  A kernel corresponding to a polynomial in Eisenstein series.
- class mul

  Product of expressions.
- class multi_iterator_counter

  The class multi_iterator_counter defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.
- class multi_iterator_counter_indv

  The class multi_iterator_counter_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.
- class multi_iterator_ordered

  The class multi_iterator_ordered defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.
- class multi_iterator_ordered_eq

  The class multi_iterator_ordered_eq defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.
- class multi_iterator_ordered_eq_indv

  The class multi_iterator_ordered_eq_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.
- class multi_iterator_permutation

  The class multi_iterator_permutation defines a multi_iterator $(i_1, i_2, ..., i_k)$, for which.
- class multi_iterator_shuffle

  The class multi_iterator_shuffle defines a multi_iterator, which runs over all shuffles of a and b.
- class multi_iterator_shuffle_prime

  The class multi_iterator_shuffle_prime defines a multi_iterator, which runs over all shuffles of a and b, excluding the first one (a,b).
- class multiple_polylog_kernel

  The integration kernel for multiple polylogarithms.
- class ncmul

  Non-commutative product of expressions.

- struct normal_map_function

  Function object to be applied by basic::normal().
- class numeric

  This class is a wrapper around CLN-numbers within the GiNaC class hierarchy.
- struct op0_is_equal
- class partition_generator

  Generate all bounded combinatorial partitions of an integer n with exactly m parts (not including zero parts) in non-decreasing order.
- class partition_with_zero_parts_generator

  Generate all bounded combinatorial partitions of an integer n with exactly m parts (including zero parts) in non-decreasing order.
- class pointer_to_map_function
- class pointer_to_map_function_1arg
- class pointer_to_map_function_2args
- class pointer_to_map_function_3args
- class pointer_to_member_to_map_function
- class pointer_to_member_to_map_function_1arg
- class pointer_to_member_to_map_function_2args
- class pointer_to_member_to_map_function_3args
- class pole_error

  Exception class thrown when a singularity is encountered.
- class possymbol

  Specialization of symbol to real positive domain.
- class power

  This class holds a two-component object, a basis and and exponent representing exponentiation.
- class print_context

  Base class for print_contexts.
- class print_context_options

  This class stores information about a registered print_context class.
- class print_csrc

  Base context for C source output.
- class print_csrc_cl_N

  Context for C source output using CLN numbers.
- class print_csrc_double

  Context for C source output using double precision.
- class print_csrc_float

  Context for C source output using float precision.
- class print_dflt

  Context for default (ginsh-parsable) output.
- class print_functor

  This class represents a print method for a certain algebraic class and print_context type.
- class print_functor_impl

  Base class for print_functor handlers.
- class print_latex

  Context for latex-parsable output.
- class print_memfun_handler

  print_functor handler for member functions of class T, context type C
- class print_options

  Flags to control the behavior of a print_context.
- class print_ptrfun_handler

  print_functor handler for pointer-to-functions of class T, context type C
- class print_python

Context for python pretty-print output.

- class print_python_repr

  Context for python-parsable output.
- class print_tree

  Context for tree-like output for debugging.
- class pseries

  This class holds a extended truncated power series (positive and negative integer powers).
- class psi1_SERIAL

  Polylogarithm and multiple polylogarithm.
- class psi2_SERIAL

  Derivatives of Psi-function (aka polygamma-functions).
- class ptr

  Class of (intrusively) reference-counted pointers that support copy-on-write semantics.
- class realsymbol

  Specialization of symbol to real domain.
- class refcounted

  Base class for reference-counted objects.
- class registered_class_options

  This class stores information about a registered GiNaC class.
- class relational

  This class holds a relation consisting of two expressions and a logical relation between them.
- class remember_strategies

  Strategies how to clean up the function remember cache.
- class remember_table

  The remember table is organized like an n-fold associative cache in a microprocessor.
- class remember_table_entry

  A single entry in the remember table of a function.
- class remember_table_list

  A list of entries in the remember table having some least significant bits of the hashvalue in common.
- struct return_type_t

  To distinguish between different kinds of non-commutative objects.
- class return_types
- class scalar_products

  Helper class for storing information about known scalar products which are to be automatically replaced by simplify_indexed().
- class series_options

  Flags to control series expansion.
- class solve_algo

  Switch to control algorithm for linear system solving.
- class spinidx

  This class holds a spinor index that can be dotted or undotted and that also has a variance.
- class spinmetric

  This class represents an antisymmetric spinor metric tensor which can be used to raise/lower indices of 2-component Weyl spinors.
- class spmapkey
- class status_flags

  Flags to store information about the state of an object.
- class structure

  Wrapper template for making GiNaC classes out of C++ structures.
- class su3d

  This class represents the tensor of symmetric su(3) structure constants.

- class su3f

  This class represents the tensor of antisymmetric su(3) structure constants.
- class su3one

  This class represents the su(3) unity element.
- class su3t

  This class represents an su(3) generator.
- class subs_options

  Flags to control the behavior of subs().
- class sy_is_less
- class sy_swap
- struct sym_desc

  This structure holds information about the highest and lowest degrees in which a symbol appears in two multivariate polynomials "a" and "b".
- class symbol

  Basic CAS symbol.
- class symbolset
- class symmetry

  This class describes the symmetry of a group of indices.
- class symminfo

  This structure stores the individual symmetrized terms obtained during the simplification of sums.
- class symminfo_is_less_by_orig
- class symminfo_is_less_by_symmterm
- class tensdelta

  This class represents the delta tensor.
- class tensepsilon

  This class represents the totally antisymmetric epsilon tensor.
- class tensmetric

  This class represents a general metric tensor which can be used to raise/lower indices.
- class tensor

  This class holds one of GiNaC's predefined special tensors such as the delta and the metric tensors.
- class terminfo

  This structure stores the original and symmetrized versions of terms obtained during the simplification of sums.
- class terminfo_is_less
- class unarchive_table_t
- class user_defined_kernel

  A user-defined integration kernel.
- class varidx

  This class holds an index with a variance (co- or contravariant).
- class visitor

  Degenerate base class for visitors.
- class wildcard

  This class acts as a wildcard for subs(), match(), has() and find().
- class zeta1_SERIAL

  Complex conjugate.
- class zeta2_SERIAL

  Alternating Euler sum or colored MZV.

Typedefs

- typedef unsigned archive_node_id

  Numerical ID value to refer to an archive_node.
- typedef unsigned archive_atom

  Numerical ID value to refer to a string.
- typedef basic ∗(∗) synthesize_func()
- typedef std::map< std::string, synthesize_func > unarchive_map_t
- typedef std::vector< ex > exvector
- typedef std::set< ex, ex_is_less > exset
- typedef std::map< ex, ex, ex_is_less > exmap
- typedef ex(∗) evalffunctype()
- typedef double(∗) FUNCP_1P(double)

  Function pointer with one function parameter.
- typedef double(∗) FUNCP_2P(double, double)

  Function pointer with two function parameters.
- typedef void(∗) FUNCP_CUBA(const int ∗, const double[ ], const int ∗, double[ ])

  Function pointer for use with the CUBA library (http://www.feynarts.de/cuba).
- typedef std::vector< expair > epvector

  expair-vector
- typedef epvector::iterator epp

  expair-vector pointer
- typedef container< std::vector > exprseq
- typedef std::multiset< unsigned > paramset
- typedef ex(∗) eval_funcp()
- typedef ex(∗) evalf_funcp()
- typedef ex(∗) conjugate_funcp()
- typedef ex(∗) real_part_funcp()
- typedef ex(∗) imag_part_funcp()
- typedef ex(∗) expand_funcp()
- typedef ex(∗) derivative_funcp()
- typedef ex(∗) expl_derivative_funcp()
- typedef ex(∗) power_funcp()
- typedef ex(∗) series_funcp()
- typedef void(∗) print_funcp()
- typedef bool(∗) info_funcp()
- typedef ex(∗) eval_funcp_1(const ex &)
- typedef ex(∗) evalf_funcp_1(const ex &)
- typedef ex(∗) conjugate_funcp_1(const ex &)
- typedef ex(∗) real_part_funcp_1(const ex &)
- typedef ex(∗) imag_part_funcp_1(const ex &)
- typedef ex(∗) expand_funcp_1(const ex &, unsigned)
- typedef ex(∗) derivative_funcp_1(const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_1(const ex &, const symbol &)
- typedef ex(∗) power_funcp_1(const ex &, const ex &)
- typedef ex(∗) series_funcp_1(const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_1(const ex &, const print_context &)
- typedef bool(∗) info_funcp_1(const ex &, unsigned)
- typedef ex(∗) eval_funcp_2(const ex &, const ex &)
- typedef ex(∗) evalf_funcp_2(const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_2(const ex &, const ex &)
- typedef ex(∗) real_part_funcp_2(const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_2(const ex &, const ex &)
- typedef ex(∗) expand_funcp_2(const ex &, const ex &, unsigned)

- typedef ex(∗) derivative_funcp_2(const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_2(const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_2(const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_2(const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_2(const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_2(const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_3(const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_3(const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_3(const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_3(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_3(const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_3(const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_3(const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_4(const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_4(const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_4(const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_4(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_4(const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_4(const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_4(const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)

- typedef ex(∗) expl_derivative_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)

- typedef ex(∗) eval_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)

- typedef ex(∗) imag_part_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)

- typedef ex(∗) power_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) evalf_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) conjugate_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) real_part_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) imag_part_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) expand_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) derivative_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) expl_derivative_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) power_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) series_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) info_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) eval_funcp_exvector(const exvector &)
- typedef ex(∗) evalf_funcp_exvector(const exvector &)
- typedef ex(∗) conjugate_funcp_exvector(const exvector &)
- typedef ex(∗) real_part_funcp_exvector(const exvector &)
- typedef ex(∗) imag_part_funcp_exvector(const exvector &)
- typedef ex(∗) expand_funcp_exvector(const exvector &, unsigned)
- typedef ex(∗) derivative_funcp_exvector(const exvector &, unsigned)
- typedef ex(∗) expl_derivative_funcp_exvector(const exvector &, const symbol &)
- typedef ex(∗) power_funcp_exvector(const exvector &, const ex &)
- typedef ex(∗) series_funcp_exvector(const exvector &, const relational &, int, unsigned)
- typedef void(∗) print_funcp_exvector(const exvector &, const print_context &)
- typedef bool(∗) info_funcp_exvector(const exvector &, unsigned)
- template<typename T, class Hash = std::hash<ex>, class KeyEqual = std::equal_to<ex>, class Allocator = std::allocator<std::↩ pair<const ex, T>>>
  using exhashmap = std::unordered_map<ex, T, Hash, KeyEqual, Allocator>
- typedef std::map< spmapkey, ex > spmap
- typedef map< error_and_integral, ex, error_and_integral_is_less > lookup_map
- typedef container< std::list > lst
- typedef std::vector< std::size_t > uintvector
- typedef std::vector< unsigned > unsignedvector
- typedef std::vector< exvector > exvectorvector

- typedef std::vector< sym_desc > sym_desc_vec
- typedef void(*) digits_changed_callback(long)

  Function pointer to implement callbacks in the case 'Digits' gets changed.
- typedef class_info< print_context_options > print_context_class_info
- typedef class_info< registered_class_options > registered_class_info

Enumerations

- enum { callback_registered = 1 }

Functions

- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (add, expairseq, print_func< print_context >(&add::do_print). print_func< print_latex >(&add::do_print_latex). print_func< print_csrc >(&add::do_print_csrc). print_↩ func< print_tree >(&add::do_print_tree). print_func< print_python_repr >(&add::do_print_python_repr)) add
- GINAC_BIND_UNARCHIVER (add)
- GINAC_DECLARE_UNARCHIVER (add)
- static void write_unsigned (std::ostream &os, unsigned val)

  Write unsigned integer quantity to stream.
- static unsigned read_unsigned (std::istream &is)

  Read unsigned integer quantity from stream.
- std::ostream & operator<< (std::ostream &os, const archive_node &n)

  Write archive_node to binary data stream.
- std::ostream & operator<< (std::ostream &os, const archive &ar)

  Write archive to binary data stream.
- std::istream & operator>> (std::istream &is, archive_node &n)

  Read archive_node from binary data stream.
- std::istream & operator>> (std::istream &is, archive &ar)

  Read archive from binary data stream.
- static synthesize_func find_factory_fcn (const std::string &name)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (basic, void, print_func< print_context >(&basic::do_print). print_func< print_tree >(&basic::do_print_tree). print_func< print_python_repr >(&basic::do_print_python_repr)) basic

  basic copy constructor: implicitly assumes that the other class is of the exact same type (as it's used by duplicate()), so it can copy the tinfo_key and the hash value.
- template<class T>
  bool is_a (const basic &obj)

  Check if obj is a T, including base classes.
- template<class T>
  bool is_exactly_a (const basic &obj)

  Check if obj is a T, not including base classes.
- template<class B, typename... Args>
  B & dynallocate (Args &&... args)

  Constructs a new (class basic or derived) B object on the heap.
- template<class B>
  B & dynallocate (std::initializer_list< ex > il)

  Constructs a new (class basic or derived) B object on the heap.
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (clifford, indexed, print_func< print_dflt >(&clifford::do_print_dflt). print_func< print_latex >(&clifford::do_print_latex). print_func< print_tree >(&clifford::do_print_tree)) GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(diracone
- print_func< print_dflt > (&diracone::do_print). print_func< print_latex >(&diracone

- GINAC_BIND_UNARCHIVER (clifford)
- GINAC_BIND_UNARCHIVER (cliffordunit)
- GINAC_BIND_UNARCHIVER (diracone)
- GINAC_BIND_UNARCHIVER (diracgamma)
- GINAC_BIND_UNARCHIVER (diracgamma5)
- GINAC_BIND_UNARCHIVER (diracgammaL)
- GINAC_BIND_UNARCHIVER (diracgammaR)
- static bool is_dirac_slash (const ex &seq0)
- static void base_and_index (const ex &c, ex &b, ex &i)

  This function decomposes gamma~mu -> (1, mu) and a\ -> (a.ix, ix).
- ex dirac_ONE (unsigned char rl=0)

  Create a Clifford unity object.
- static unsigned get_dim_uint (const ex &e)
- ex clifford_unit (const ex &mu, const ex &metr, unsigned char rl=0)

  Create a Clifford unit object.
- ex dirac_gamma (const ex &mu, unsigned char rl=0)

  Create a Dirac gamma object.
- ex dirac_gamma5 (unsigned char rl=0)

  Create a Dirac gamma5 object.
- ex dirac_gammaL (unsigned char rl=0)

  Create a Dirac gammaL object.
- ex dirac_gammaR (unsigned char rl=0)

  Create a Dirac gammaR object.
- ex dirac_slash (const ex &e, const ex &dim, unsigned char rl=0)

  *Create a term of the form e_mu * gamma~mu with a unique index mu.*
- static unsigned char get_representation_label (const return_type_t &ti)

  Extract representation label from tinfo key (as returned by return_type_tinfo()).
- static ex trace_string (exvector::const_iterator ix, size_t num)

  Take trace of a string of an even number of Dirac gammas given a vector of indices.
- ex dirac_trace (const ex &e, const std::set< unsigned char > &rls, const ex &trONE=4)

  Calculate dirac traces over the specified set of representation labels.
- ex dirac_trace (const ex &e, const lst &rll, const ex &trONE=4)

  Calculate dirac traces over the specified list of representation labels.
- ex dirac_trace (const ex &e, unsigned char rl=0, const ex &trONE=4)

  Calculate the trace of an expression containing gamma objects with a specified representation label.
- ex canonicalize_clifford (const ex &e)

  Bring all products of clifford objects in an expression into a canonical order.
- ex clifford_star_bar (const ex &e, bool do_bar, unsigned options)

  An auxillary function performing clifford_star() and clifford_bar().
- ex clifford_prime (const ex &e)

  Automorphism of the Clifford algebra, simply changes signs of all clifford units.
- ex remove_dirac_ONE (const ex &e, unsigned char rl=0, unsigned options=0)

  Replaces dirac_ONE's (with a representation_label no less than rl) in e with 1.
- int clifford_max_label (const ex &e, bool ignore_ONE=false)

  Returns the maximal representation label of a clifford object if e contains at least one, otherwise returns -1.
- ex clifford_norm (const ex &e)

  Calculation of the norm in the Clifford algebra.
- ex clifford_inverse (const ex &e)

  Calculation of the inverse in the Clifford algebra.
- ex lst_to_clifford (const ex &v, const ex &mu, const ex &metr, unsigned char rl=0)

  List or vector conversion into the Clifford vector.

- ex lst_to_clifford (const ex &v, const ex &e)

  List or vector conversion into the Clifford vector.
- static ex get_clifford_comp (const ex &e, const ex &c, bool root=true)

  Auxiliary structure to define a function for striping one Clifford unit from vectors.
- lst clifford_to_lst (const ex &e, const ex &c, bool algebraic=true)

  An inverse function to lst_to_clifford().
- ex clifford_moebius_map (const ex &a, const ex &b, const ex &c, const ex &d, const ex &v, const ex &G, unsigned char rl=0)

  Calculations of Moebius transformations (conformal map) defined by a 2x2 Clifford matrix (a b\c d) in linear spaces with arbitrary signature.
- ex clifford_moebius_map (const ex &M, const ex &v, const ex &G, unsigned char rl=0)

  The second form of Moebius transformations defined by a 2x2 Clifford matrix M This function takes the transformation matrix M as a single entity.
- GINAC_DECLARE_UNARCHIVER (clifford)
- GINAC_DECLARE_UNARCHIVER (diracone)
- GINAC_DECLARE_UNARCHIVER (cliffordunit)
- GINAC_DECLARE_UNARCHIVER (diracgamma)
- GINAC_DECLARE_UNARCHIVER (diracgamma5)
- GINAC_DECLARE_UNARCHIVER (diracgammaL)
- GINAC_DECLARE_UNARCHIVER (diracgammaR)
- bool is_clifford_tinfo (const return_type_t &ti)

  Check whether a given return_type_t object (as returned by return_type_tinfo() is that of a clifford object (with an arbitrary representation label).
- ex clifford_bar (const ex &e)

  Main anti-automorphism of the Clifford algebra: makes reversion and changes signs of all clifford units.
- ex clifford_star (const ex &e)

  Reversion of the Clifford algebra, reverse the order of all clifford units in ncmul.
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (su3one, tensor, print_func< print_dflt >(&su3one::do_print). print_func< print_latex >(&su3one::do_print_latex)) GINAC_IMPLEMENT_REGISTERED_CLASS_↩ OPT(su3t
- print_func< print_dflt > (&su3t::do_print). print_func< print_latex >(&su3t
- GINAC_BIND_UNARCHIVER (color)
- GINAC_BIND_UNARCHIVER (su3one)
- GINAC_BIND_UNARCHIVER (su3t)
- GINAC_BIND_UNARCHIVER (su3f)
- GINAC_BIND_UNARCHIVER (su3d)
- static ex permute_free_index_to_front (const exvector &iv3, const exvector &iv2, int &sig)

  Given a vector iv3 of three indices and a vector iv2 of two indices that is a subset of iv3, return the (free) index that is in iv3 but not in iv2 and the sign introduced by permuting that index to the front.
- ex color_ONE (unsigned char rl=0)

  Create the su(3) unity element.
- ex color_T (const ex &a, unsigned char rl=0)

  Create an su(3) generator.
- ex color_f (const ex &a, const ex &b, const ex &c)

  Create an su(3) antisymmetric structure constant.
- ex color_d (const ex &a, const ex &b, const ex &c)

  Create an su(3) symmetric structure constant.
- ex color_h (const ex &a, const ex &b, const ex &c)

  This returns the linear combination d.a.b.c+I∗f.a.b.c.
- static bool is_color_tinfo (const return_type_t &ti)

  Check whether a given tinfo key (as returned by return_type_tinfo() is that of a color object (with an arbitrary representation label).
- static unsigned char get_representation_label (const return_type_t &ti)

Extract representation label from tinfo key (as returned by return_type_tinfo()).

- ex color_trace (const ex &e, const std::set< unsigned char > &rls)

  Calculate color traces over the specified set of representation labels.
- ex color_trace (const ex &e, const lst &rll)

  Calculate color traces over the specified list of representation labels.
- ex color_trace (const ex &e, unsigned char rl=0)

  Calculate the trace of an expression containing color objects with a specified representation label.
- GINAC_DECLARE_UNARCHIVER (color)
- GINAC_DECLARE_UNARCHIVER (su3one)
- GINAC_DECLARE_UNARCHIVER (su3t)
- GINAC_DECLARE_UNARCHIVER (su3f)
- GINAC_DECLARE_UNARCHIVER (su3d)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (constant, basic, print_func< print_context >(&constant::do_print). print_func< print_latex >(&constant::do_print_latex). print_func< print_tree >(&constant::do_print_tree). print_func< print_python_repr >(&constant::do_print_python_repr)) const ant
- GINAC_BIND_UNARCHIVER (constant)
- GINAC_DECLARE_UNARCHIVER (constant)
- static unsigned crc32 (const char *c, const unsigned len, const unsigned crcinit)
- bool are_ex_trivially_equal (const ex &e1, const ex &e2)

  Compare two objects of class quickly without doing a deep tree traversal.
- std::ostream & operator<< (std::ostream &os, const exvector &e)
- std::ostream & operator<< (std::ostream &os, const exset &e)
- std::ostream & operator<< (std::ostream &os, const exmap &e)
- size_t nops (const ex &thisex)
- ex expand (const ex &thisex, unsigned options=0)
- ex conjugate (const ex &thisex)
- ex real_part (const ex &thisex)
- ex imag_part (const ex &thisex)
- bool has (const ex &thisex, const ex &pattern, unsigned options=0)
- bool find (const ex &thisex, const ex &pattern, exset &found)
- bool is_polynomial (const ex &thisex, const ex &vars)
- int degree (const ex &thisex, const ex &s)
- int ldegree (const ex &thisex, const ex &s)
- ex coeff (const ex &thisex, const ex &s, int n=1)
- ex numer (const ex &thisex)
- ex denom (const ex &thisex)
- ex numer_denom (const ex &thisex)
- ex normal (const ex &thisex)
- ex to_rational (const ex &thisex, exmap &repl)
- ex to_polynomial (const ex &thisex, exmap &repl)
- ex collect (const ex &thisex, const ex &s, bool distributed=false)
- ex eval (const ex &thisex)
- ex evalf (const ex &thisex)
- ex evalm (const ex &thisex)
- ex eval_integ (const ex &thisex)
- ex diff (const ex &thisex, const symbol &s, unsigned nth=1)
- ex series (const ex &thisex, const ex &r, int order, unsigned options=0)
- bool match (const ex &thisex, const ex &pattern, exmap &repl_lst)
- ex simplify_indexed (const ex &thisex, unsigned options=0)
- ex simplify_indexed (const ex &thisex, const scalar_products &sp, unsigned options=0)
- ex symmetrize (const ex &thisex)
- ex symmetrize (const ex &thisex, const lst &l)
- ex antisymmetrize (const ex &thisex)
- ex antisymmetrize (const ex &thisex, const lst &l)

- ex symmetrize_cyclic (const ex &thisex)
- ex symmetrize_cyclic (const ex &thisex, const lst &l)
- ex op (const ex &thisex, size_t i)
- ex lhs (const ex &thisex)
- ex rhs (const ex &thisex)
- bool is_zero (const ex &thisex)
- void swap (ex &e1, ex &e2)
- ex subs (const ex &thisex, const exmap &m, unsigned options=0)
- ex subs (const ex &thisex, const lst &ls, const lst &lr, unsigned options=0)
- ex subs (const ex &thisex, const ex &e, unsigned options=0)
- template<class T>
  bool is_a (const ex &obj)

  Check if ex is a handle to a T, including base classes.
- template<class T>
  bool is_exactly_a (const ex &obj)

  Check if ex is a handle to a T, not including base classes.
- template<class T>
  attribute_pure const T & ex_to (const ex &e)

  Return a reference to the basic-derived class T object embedded in an expression.
- void compile_ex (const ex &expr, const symbol &sym, FUNCP_1P &fp, const std::string filename="")

  Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.
- void compile_ex (const ex &expr, const symbol &sym1, const symbol &sym2, FUNCP_2P &fp, const std::↩ string filename="")

  Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.
- void compile_ex (const lst &exprs, const lst &syms, FUNCP_CUBA &fp, const std::string filename="")

  Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.
- void link_ex (const std::string filename, FUNCP_1P &fp)

  Opens an existing so-file and returns a function pointer of type FUNCP_1P to the contained function.
- void link_ex (const std::string filename, FUNCP_CUBA &fp)

  Opens an existing so-file and returns a function pointer of type FUNCP_CUBA to the contained function.
- void unlink_ex (const std::string filename)

  Closes all linked .so files that have the supplied filename.
- void swap (expair &e1, expair &e2)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (expairseq, basic, print_func< print_context >(&expairseq::do_print). print_func< print_tree >(&expairseq::do_print_tree)) class epp_is_less
- epvector * conjugateepvector (const epvector &)

  Complex conjugate every element of an epvector.
- ex factor (const ex &poly, unsigned options)

  Interface function to the outside world.
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (fail, basic, print_func< print_context >(&fail::do_print). print_func< print_tree >(&fail::do_print_tree)) GINAC_BIND_UNARCHIVER(fail)
- GINAC_DECLARE_UNARCHIVER (fail)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (fderivative, function, print_func< print_context >(&fderivative::do_print). print_func< print_latex >(&fderivative::do_print_latex). print_func< print_csrc >(&fderivative::do_print_csrc). print_func< print_tree >(&fderivative::do_print_tree)) fderivative
- GINAC_BIND_UNARCHIVER (fderivative)
- GINAC_DECLARE_UNARCHIVER (fderivative)
- GINAC_BIND_UNARCHIVER (function)
- GINAC_DECLARE_UNARCHIVER (function)
- template<typename T>
  bool is_the_function (const ex &x)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Gt, basic, print_func< print_dflt >(&Gt::do_print). print_func< print_python >(&Gt::do_print). print_func< print_latex >(&Gt::do_print_latex)) Gt

- GINAC_BIND_UNARCHIVER (Gt)
- GINAC_DECLARE_UNARCHIVER (Gt)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (idx, basic, print_func< print_context >(&idx::do_print). print_func< print_latex >(&idx::do_print_latex). print_func< print_csrc >(&idx::do_print_csrc). print_func< print_tree >(&idx::do_print_tree)) GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(varidx
- print_func< print_context > (&varidx::do_print). print_func< print_latex >(&varidx
- GINAC_BIND_UNARCHIVER (idx)
- GINAC_BIND_UNARCHIVER (varidx)
- GINAC_BIND_UNARCHIVER (spinidx)
- bool is_dummy_pair (const idx &i1, const idx &i2)

  Check whether two indices form a dummy pair.
- bool is_dummy_pair (const ex &e1, const ex &e2)

  Check whether two expressions form a dummy index pair.
- void find_free_and_dummy (exvector::const_iterator it, exvector::const_iterator itend, exvector &out_free, exvector &out_dummy)

  Given a vector of indices, split them into two vectors, one containing the free indices, the other containing the dummy indices (numeric indices are neither free nor dummy ones).
- ex minimal_dim (const ex &dim1, const ex &dim2)

  Return the minimum of two index dimensions.
- GINAC_DECLARE_UNARCHIVER (idx)
- GINAC_DECLARE_UNARCHIVER (varidx)
- GINAC_DECLARE_UNARCHIVER (spinidx)
- void find_free_and_dummy (const exvector &v, exvector &out_free, exvector &out_dummy)

  Given a vector of indices, split them into two vectors, one containing the free indices, the other containing the dummy indices (numeric indices are neither free nor dummy ones).
- void find_dummy_indices (const exvector &v, exvector &out_dummy)

  Given a vector of indices, find the dummy indices.
- size_t count_dummy_indices (const exvector &v)

  Count the number of dummy index pairs in an index vector.
- size_t count_free_indices (const exvector &v)

  Count the number of dummy index pairs in an index vector.
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (indexed, exprseq, print_func< print_context >(&indexed::do_print). print_func< print_latex >(&indexed::do_print_latex). print_func< print_tree >(&indexed::do_print_tree)) indexed
- GINAC_BIND_UNARCHIVER (indexed)
- static bool indices_consistent (const exvector &v1, const exvector &v2)

  Check whether two sorted index vectors are consistent (i.e.
- template<class T>
  size_t number_of_type (const exvector &v)
- template<class T>
  static ex rename_dummy_indices (const ex &e, exvector &global_dummy_indices, exvector &local_dummy↩_indices)

  Rename dummy indices in an expression.
- static void find_variant_indices (const exvector &v, exvector &variant_indices)

  Given a set of indices, extract those of class varidx.
- bool reposition_dummy_indices (ex &e, exvector &variant_dummy_indices, exvector &moved_indices)

  Raise/lower dummy indices in a single indexed objects to canonicalize their variance.
- static void product_to_exvector (const ex &e, exvector &v, bool &non_commutative)
- template<class T>
  ex idx_symmetrization (const ex &r, const exvector &local_dummy_indices)
- ex simplify_indexed (const ex &e, exvector &free_indices, exvector &dummy_indices, const scalar_products &sp)

  Simplify indexed expression, return list of free indices.

- ex simplify_indexed_product (const ex &e, exvector &free_indices, exvector &dummy_indices, const scalar_products &sp)

  Simplify product of indexed expressions (commutative, noncommutative and simple squares), return list of free indices.
- bool hasindex (const ex &x, const ex &sym)
- exvector get_all_dummy_indices_safely (const ex &e)

  More reliable version of the form.
- exvector get_all_dummy_indices (const ex &e)

  Returns all dummy indices from the exvector.
- lst rename_dummy_indices_uniquely (const exvector &va, const exvector &vb)

  Similar to above, where va and vb are the same and the return value is a list of two lists for substitution in b.
- ex rename_dummy_indices_uniquely (const exvector &va, const exvector &vb, const ex &b)

  Same as above, where va and vb contain the indices of a and b and are sorted.
- ex rename_dummy_indices_uniquely (const ex &a, const ex &b)

  Returns b with all dummy indices, which are common with a, renamed.
- ex rename_dummy_indices_uniquely (exvector &va, const ex &b, bool modify_va=false)

  Returns b with all dummy indices, which are listed in va, renamed if modify_va is set to TRUE all dummy indices of b will be appended to va.
- ex expand_dummy_sum (const ex &e, bool subs_idx=false)

  This function returns the given expression with expanded sums for all dummy index summations, where the dimensionality of the dummy index is a nonnegative integer.
- GINAC_DECLARE_UNARCHIVER (indexed)
- static ex conjugate_evalf (const ex &arg)
- static ex conjugate_eval (const ex &arg)
- static void conjugate_print_latex (const ex &arg, const print_context &c)
- static ex conjugate_conjugate (const ex &arg)
- static ex conjugate_expl_derivative (const ex &arg, const symbol &s)
- static ex conjugate_real_part (const ex &arg)
- static ex conjugate_imag_part (const ex &arg)
- static bool func_arg_info (const ex &arg, unsigned inf)
- static bool conjugate_info (const ex &arg, unsigned inf)
- REGISTER_FUNCTION (conjugate_function, eval_func(conjugate_eval). evalf_func(conjugate_evalf). expl_derivative_func(conjugate_expl_derivative). info_func(conjugate_info). print_func< print_latex >(conjugate_print_latex). conjugate_func(conjugate_conjugate). real_part_func(conjugate_real_part). imag_part_func(conjugate_imag_part). set_name("conjugate","conjugate"))
- static ex real_part_evalf (const ex &arg)
- static ex real_part_eval (const ex &arg)
- static void real_part_print_latex (const ex &arg, const print_context &c)
- static ex real_part_conjugate (const ex &arg)
- static ex real_part_real_part (const ex &arg)
- static ex real_part_imag_part (const ex &arg)
- static ex real_part_expl_derivative (const ex &arg, const symbol &s)
- REGISTER_FUNCTION (real_part_function, eval_func(real_part_eval). evalf_func(real_part_evalf). expl↩_derivative_func(real_part_expl_derivative). print_func< print_latex >(real_part_print_latex). conjugate↩_func(real_part_conjugate). real_part_func(real_part_real_part). imag_part_func(real_part_imag_part). set_name("real_part","real_part"))
- static ex imag_part_evalf (const ex &arg)
- static ex imag_part_eval (const ex &arg)
- static void imag_part_print_latex (const ex &arg, const print_context &c)
- static ex imag_part_conjugate (const ex &arg)
- static ex imag_part_real_part (const ex &arg)
- static ex imag_part_imag_part (const ex &arg)
- static ex imag_part_expl_derivative (const ex &arg, const symbol &s)

- REGISTER_FUNCTION (imag_part_function, eval_func(imag_part_eval). evalf_func(imag_part_evalf). expl_derivative_func(imag_part_expl_derivative). print_func< print_latex >(imag_part_print_latex). conjugate_func(imag_part_conjugate). real_part_func(imag_part_real_part). imag_part_func(imag_part_imag_part). set_name("imag_part","imag_part"))
- static ex abs_evalf (const ex &arg)
- static ex abs_eval (const ex &arg)
- static ex abs_expand (const ex &arg, unsigned options)
- static ex abs_expl_derivative (const ex &arg, const symbol &s)
- static void abs_print_latex (const ex &arg, const print_context &c)
- static void abs_print_csrc_float (const ex &arg, const print_context &c)
- static ex abs_conjugate (const ex &arg)
- static ex abs_real_part (const ex &arg)
- static ex abs_imag_part (const ex &arg)
- static ex abs_power (const ex &arg, const ex &exp)
- bool abs_info (const ex &arg, unsigned inf)
- REGISTER_FUNCTION (abs, eval_func(abs_eval). evalf_func(abs_evalf). expand_func(abs_expand). expl_derivative_func(abs_expl_derivative). info_func(abs_info). print_func< print_latex >(abs_print_latex). print_func< print_csrc_float >(abs_print_csrc_float). print_func< print_csrc_double >(abs_print_csrc_float). conjugate_func(abs_conjugate). real_part_func(abs_real_part). imag_part_func(abs_imag_part). power_↩ func(abs_power))
- static ex step_evalf (const ex &arg)
- static ex step_eval (const ex &arg)
- static ex step_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex step_conjugate (const ex &arg)
- static ex step_real_part (const ex &arg)
- static ex step_imag_part (const ex &arg)
- REGISTER_FUNCTION (step, eval_func(step_eval). evalf_func(step_evalf). series_func(step_series). conjugate_func(step_conjugate). real_part_func(step_real_part). imag_part_func(step_imag_part))
- static ex csgn_evalf (const ex &arg)
- static ex csgn_eval (const ex &arg)
- static ex csgn_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex csgn_conjugate (const ex &arg)
- static ex csgn_real_part (const ex &arg)
- static ex csgn_imag_part (const ex &arg)
- static ex csgn_power (const ex &arg, const ex &exp)
- REGISTER_FUNCTION (csgn, eval_func(csgn_eval). evalf_func(csgn_evalf). series_func(csgn_series). conjugate_func(csgn_conjugate). real_part_func(csgn_real_part). imag_part_func(csgn_imag_part). power_func(csgn_power))
- static ex eta_evalf (const ex &x, const ex &y)
- static ex eta_eval (const ex &x, const ex &y)
- static ex eta_series (const ex &x, const ex &y, const relational &rel, int order, unsigned options)
- static ex eta_conjugate (const ex &x, const ex &y)
- static ex eta_real_part (const ex &x, const ex &y)
- static ex eta_imag_part (const ex &x, const ex &y)
- REGISTER_FUNCTION (eta, eval_func(eta_eval). evalf_func(eta_evalf). series_func(eta_series). latex_name("\\eta"). set_symmetry(sy_symm(0, 1)). conjugate_func(eta_conjugate). real_part_↩ func(eta_real_part). imag_part_func(eta_imag_part))
- static ex Li2_evalf (const ex &x)
- static ex Li2_eval (const ex &x)
- static ex Li2_deriv (const ex &x, unsigned deriv_param)
- static ex Li2_series (const ex &x, const relational &rel, int order, unsigned options)
- static ex Li2_conjugate (const ex &x)
- REGISTER_FUNCTION (Li2, eval_func(Li2_eval). evalf_func(Li2_evalf). derivative_func(Li2_deriv). series↩ _func(Li2_series). conjugate_func(Li2_conjugate). latex_name("\\mathrm{Li}_2"))
- static ex Li3_eval (const ex &x)

- REGISTER_FUNCTION (Li3, eval_func(Li3_eval). latex_name("\\mathrm{Li}_3"))
- static ex zetaderiv_eval (const ex &n, const ex &x)
- static ex zetaderiv_deriv (const ex &n, const ex &x, unsigned deriv_param)
- REGISTER_FUNCTION (zetaderiv, eval_func(zetaderiv_eval). derivative_func(zetaderiv_deriv). latex_↩ name("\\zeta^\\prime"))
- static ex factorial_evalf (const ex &x)
- static ex factorial_eval (const ex &x)
- static void factorial_print_dflt_latex (const ex &x, const print_context &c)
- static ex factorial_conjugate (const ex &x)
- static ex factorial_real_part (const ex &x)
- static ex factorial_imag_part (const ex &x)
- REGISTER_FUNCTION (factorial, eval_func(factorial_eval). evalf_func(factorial_evalf). print_func< print_dflt >(factorial_print_dflt_latex). print_func< print_latex >(factorial_print_dflt_latex). conjugate_↩ func(factorial_conjugate). real_part_func(factorial_real_part). imag_part_func(factorial_imag_part))
- static ex binomial_evalf (const ex &x, const ex &y)
- static ex binomial_sym (const ex &x, const numeric &y)
- static ex binomial_eval (const ex &x, const ex &y)
- static ex binomial_conjugate (const ex &x, const ex &y)
- static ex binomial_real_part (const ex &x, const ex &y)
- static ex binomial_imag_part (const ex &x, const ex &y)
- REGISTER_FUNCTION (binomial, eval_func(binomial_eval). evalf_func(binomial_evalf). conjugate_↩ func(binomial_conjugate). real_part_func(binomial_real_part). imag_part_func(binomial_imag_part))
- static ex Order_eval (const ex &x)
- static ex Order_series (const ex &x, const relational &r, int order, unsigned options)
- static ex Order_conjugate (const ex &x)
- static ex Order_real_part (const ex &x)
- static ex Order_imag_part (const ex &x)
- static ex Order_power (const ex &x, const ex &e)
- static ex Order_expl_derivative (const ex &arg, const symbol &s)
- REGISTER_FUNCTION (Order, eval_func(Order_eval). series_func(Order_series). latex_name("\\mathcal{O}"). expl_derivative_func(Order_expl_derivative). conjugate_func(Order_conjugate). real_part_func(Order_real_part). imag_part_func(Order_imag_part). power_func(Order_power))
- ex lsolve (const ex &eqns, const ex &symbols, unsigned options=solve_algo::automatic)

  Factorial function.
- const numeric fsolve (const ex &f, const symbol &x, const numeric &x1, const numeric &x2)

  Find a real root of real-valued function f(x) numerically within a given interval.
- template<typename T1>
  function zeta (const T1 &p1)
- template<typename T1, typename T2>
  function zeta (const T1 &p1, const T2 &p2)
- template<> bool is_the_function< zeta_SERIAL > (const ex &x)
- template<typename T1, typename T2>
  function G (const T1 &x, const T2 &y)
- template<typename T1, typename T2, typename T3>
  function G (const T1 &x, const T2 &s, const T3 &y)
- template<> bool is_the_function< G_SERIAL > (const ex &x)
- template<typename T1>
  function psi (const T1 &p1)
- template<typename T1, typename T2>
  function psi (const T1 &p1, const T2 &p2)
- template<> bool is_the_function< psi_SERIAL > (const ex &x)
- template<typename T1, typename T2>
  function iterated_integral (const T1 &kernel_lst, const T2 &lambda)
- template<typename T1, typename T2, typename T3>
  function iterated_integral (const T1 &kernel_lst, const T2 &lambda, const T3 &N_trunc)

- template<> bool is_the_function< iterated_integral_SERIAL > (const ex &x)
- bool is_order_function (const ex &e)

  Check whether a function is the Order (O(n)) function.
- ex convert_H_to_Li (const ex &parameterlst, const ex &arg)

  Converts a given list containing parameters for H in Remiddi/Vermaseren notation into the corresponding GiNaC functions.
- static ex EllipticK_evalf (const ex &k)
- static ex EllipticK_eval (const ex &k)
- static ex EllipticK_deriv (const ex &k, unsigned deriv_param)
- static ex EllipticK_series (const ex &k, const relational &rel, int order, unsigned options)
- static void EllipticK_print_latex (const ex &k, const print_context &c)
- REGISTER_FUNCTION (EllipticK, evalf_func(EllipticK_evalf). eval_func(EllipticK_eval). derivative_func(EllipticK_deriv). series_func(EllipticK_series). print_func< print_latex >(EllipticK_print_latex). do_not_evalf_params())
- static ex EllipticE_evalf (const ex &k)
- static ex EllipticE_eval (const ex &k)
- static ex EllipticE_deriv (const ex &k, unsigned deriv_param)
- static ex EllipticE_series (const ex &k, const relational &rel, int order, unsigned options)
- static void EllipticE_print_latex (const ex &k, const print_context &c)
- REGISTER_FUNCTION (EllipticE, evalf_func(EllipticE_evalf). eval_func(EllipticE_eval). derivative_func(EllipticE_deriv). series_func(EllipticE_series). print_func< print_latex >(EllipticE_print_latex). do_not_evalf_params())
- static ex iterated_integral_evalf_impl (const ex &kernel_lst, const ex &lambda, const ex &N_trunc)
- static ex iterated_integral2_evalf (const ex &kernel_lst, const ex &lambda)
- static ex iterated_integral3_evalf (const ex &kernel_lst, const ex &lambda, const ex &N_trunc)
- static ex iterated_integral2_eval (const ex &kernel_lst, const ex &lambda)
- static ex iterated_integral3_eval (const ex &kernel_lst, const ex &lambda, const ex &N_trunc)
- static ex lgamma_evalf (const ex &x)
- static ex lgamma_eval (const ex &x)

  Evaluation of lgamma(x), the natural logarithm of the Gamma function.
- static ex lgamma_deriv (const ex &x, unsigned deriv_param)
- static ex lgamma_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex lgamma_conjugate (const ex &x)
- REGISTER_FUNCTION (lgamma, eval_func(lgamma_eval). evalf_func(lgamma_evalf). derivative_func(lgamma_deriv). series_func(lgamma_series). conjugate_func(lgamma_conjugate). latex_name("\\log \\Gamma"))
- static ex tgamma_evalf (const ex &x)
- static ex tgamma_eval (const ex &x)

  Evaluation of tgamma(x), the true Gamma function.
- static ex tgamma_deriv (const ex &x, unsigned deriv_param)
- static ex tgamma_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex tgamma_conjugate (const ex &x)
- REGISTER_FUNCTION (tgamma, eval_func(tgamma_eval). evalf_func(tgamma_evalf). derivative_func(tgamma_deriv). series_func(tgamma_series). conjugate_func(tgamma_conjugate). latex_name("\\Gamma"))
- static ex beta_evalf (const ex &x, const ex &y)
- static ex beta_eval (const ex &x, const ex &y)
- static ex beta_deriv (const ex &x, const ex &y, unsigned deriv_param)
- static ex beta_series (const ex &arg1, const ex &arg2, const relational &rel, int order, unsigned options)
- REGISTER_FUNCTION (beta, eval_func(beta_eval). evalf_func(beta_evalf). derivative_func(beta_deriv). series_func(beta_series). latex_name("\\mathrm{B}"). set_symmetry(sy_symm(0, 1)))
- static ex psi1_evalf (const ex &x)
- static ex psi1_eval (const ex &x)

  Evaluation of digamma-function psi(x).

- static ex psi1_deriv (const ex &x, unsigned deriv_param)
- static ex psi1_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex psi2_evalf (const ex &n, const ex &x)
- static ex psi2_eval (const ex &n, const ex &x)

  Evaluation of polygamma-function psi(n,x).
- static ex psi2_deriv (const ex &n, const ex &x, unsigned deriv_param)
- static ex psi2_series (const ex &n, const ex &arg, const relational &rel, int order, unsigned options)
- static ex G2_evalf (const ex &x_, const ex &y)
- static ex G2_eval (const ex &x_, const ex &y)
- static ex G3_evalf (const ex &x_, const ex &s_, const ex &y)
- static ex G3_eval (const ex &x_, const ex &s_, const ex &y)
- static ex Li_evalf (const ex &m_, const ex &x_)
- static ex Li_eval (const ex &m_, const ex &x_)
- static ex Li_series (const ex &m, const ex &x, const relational &rel, int order, unsigned options)
- static ex Li_deriv (const ex &m_, const ex &x_, unsigned deriv_param)
- static void Li_print_latex (const ex &m_, const ex &x_, const print_context &c)
- REGISTER_FUNCTION (Li, evalf_func(Li_evalf). eval_func(Li_eval). series_func(Li_series). derivative_↩ func(Li_deriv). print_func< print_latex >(Li_print_latex). do_not_evalf_params())
- static ex S_evalf (const ex &n, const ex &p, const ex &x)
- static ex S_eval (const ex &n, const ex &p, const ex &x)
- static ex S_series (const ex &n, const ex &p, const ex &x, const relational &rel, int order, unsigned options)
- static ex S_deriv (const ex &n, const ex &p, const ex &x, unsigned deriv_param)
- static void S_print_latex (const ex &n, const ex &p, const ex &x, const print_context &c)
- REGISTER_FUNCTION (S, evalf_func(S_evalf). eval_func(S_eval). series_func(S_series). derivative_↩ func(S_deriv). print_func< print_latex >(S_print_latex). do_not_evalf_params())
- static ex H_evalf (const ex &x1, const ex &x2)
- static ex H_eval (const ex &m_, const ex &x)
- static ex H_series (const ex &m, const ex &x, const relational &rel, int order, unsigned options)
- static ex H_deriv (const ex &m_, const ex &x, unsigned deriv_param)
- static void H_print_latex (const ex &m_, const ex &x, const print_context &c)
- REGISTER_FUNCTION (H, evalf_func(H_evalf). eval_func(H_eval). series_func(H_series). derivative_↩ func(H_deriv). print_func< print_latex >(H_print_latex). do_not_evalf_params())
- static ex zeta1_evalf (const ex &x)
- static ex zeta1_eval (const ex &m)
- static ex zeta1_deriv (const ex &m, unsigned deriv_param)
- static void zeta1_print_latex (const ex &m_, const print_context &c)
- static ex zeta2_evalf (const ex &x, const ex &s)
- static ex zeta2_eval (const ex &m, const ex &s_)
- static ex zeta2_deriv (const ex &m, const ex &s, unsigned deriv_param)
- static void zeta2_print_latex (const ex &m_, const ex &s_, const print_context &c)
- static ex exp_evalf (const ex &x)
- static ex exp_eval (const ex &x)
- static ex exp_expand (const ex &arg, unsigned options)
- static ex exp_deriv (const ex &x, unsigned deriv_param)
- static ex exp_real_part (const ex &x)
- static ex exp_imag_part (const ex &x)
- static ex exp_conjugate (const ex &x)
- static ex exp_power (const ex &x, const ex &a)
- static bool exp_info (const ex &x, unsigned inf)
- REGISTER_FUNCTION (exp, eval_func(exp_eval). evalf_func(exp_evalf). info_func(exp_info). expand_↩ func(exp_expand). derivative_func(exp_deriv). real_part_func(exp_real_part). imag_part_func(exp_imag_part). conjugate_func(exp_conjugate). power_func(exp_power). latex_name("\\exp"))
- static ex log_evalf (const ex &x)
- static ex log_eval (const ex &x)
- static ex log_deriv (const ex &x, unsigned deriv_param)

- static ex log_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex log_real_part (const ex &x)
- static ex log_imag_part (const ex &x)
- static ex log_expand (const ex &arg, unsigned options)
- static ex log_conjugate (const ex &x)
- static bool log_info (const ex &x, unsigned inf)
- REGISTER_FUNCTION (log, eval_func(log_eval). evalf_func(log_evalf). info_func(log_info). expand↩ _func(log_expand). derivative_func(log_deriv). series_func(log_series). real_part_func(log_real_part). imag_part_func(log_imag_part). conjugate_func(log_conjugate). latex_name("\\ln"))
- static ex sin_evalf (const ex &x)
- static ex sin_eval (const ex &x)
- static ex sin_deriv (const ex &x, unsigned deriv_param)
- static ex sin_real_part (const ex &x)
- static ex sin_imag_part (const ex &x)
- static ex sin_conjugate (const ex &x)
- static bool trig_info (const ex &x, unsigned inf)
- REGISTER_FUNCTION (sin, eval_func(sin_eval). evalf_func(sin_evalf). info_func(trig_info). derivative↩ _func(sin_deriv). real_part_func(sin_real_part). imag_part_func(sin_imag_part). conjugate_↩ func(sin_conjugate). latex_name("\\sin"))
- static ex cos_evalf (const ex &x)
- static ex cos_eval (const ex &x)
- static ex cos_deriv (const ex &x, unsigned deriv_param)
- static ex cos_real_part (const ex &x)
- static ex cos_imag_part (const ex &x)
- static ex cos_conjugate (const ex &x)
- REGISTER_FUNCTION (cos, eval_func(cos_eval). info_func(trig_info). evalf_func(cos_evalf). derivative↩ _func(cos_deriv). real_part_func(cos_real_part). imag_part_func(cos_imag_part). conjugate_↩ func(cos_conjugate). latex_name("\\cos"))
- static ex tan_evalf (const ex &x)
- static ex tan_eval (const ex &x)
- static ex tan_deriv (const ex &x, unsigned deriv_param)
- static ex tan_real_part (const ex &x)
- static ex tan_imag_part (const ex &x)
- static ex tan_series (const ex &x, const relational &rel, int order, unsigned options)
- static ex tan_conjugate (const ex &x)
- REGISTER_FUNCTION (tan, eval_func(tan_eval). evalf_func(tan_evalf). info_func(trig_info). derivative↩ _func(tan_deriv). series_func(tan_series). real_part_func(tan_real_part). imag_part_func(tan_imag_part). conjugate_func(tan_conjugate). latex_name("\\tan"))
- static ex asin_evalf (const ex &x)
- static ex asin_eval (const ex &x)
- static ex asin_deriv (const ex &x, unsigned deriv_param)
- static ex asin_conjugate (const ex &x)
- static bool asin_info (const ex &x, unsigned inf)
- REGISTER_FUNCTION (asin, eval_func(asin_eval). evalf_func(asin_evalf). info_func(asin_info). derivative_func(asin_deriv). conjugate_func(asin_conjugate). latex_name("\\arcsin"))
- static ex acos_evalf (const ex &x)
- static ex acos_eval (const ex &x)
- static ex acos_deriv (const ex &x, unsigned deriv_param)
- static ex acos_conjugate (const ex &x)
- REGISTER_FUNCTION (acos, eval_func(acos_eval). evalf_func(acos_evalf). info_func(asin_info). derivative_func(acos_deriv). conjugate_func(acos_conjugate). latex_name("\\arccos"))
- static ex atan_evalf (const ex &x)
- static ex atan_eval (const ex &x)
- static ex atan_deriv (const ex &x, unsigned deriv_param)
- static ex atan_series (const ex &arg, const relational &rel, int order, unsigned options)

- static ex atan_conjugate (const ex &x)
- static bool atan_info (const ex &x, unsigned inf)
- REGISTER_FUNCTION (atan, eval_func(atan_eval). evalf_func(atan_evalf). info_func(atan_info). derivative_func(atan_deriv). series_func(atan_series). conjugate_func(atan_conjugate). latex_name("\\arctan"))
- static ex atan2_evalf (const ex &y, const ex &x)
- static ex atan2_eval (const ex &y, const ex &x)
- static ex atan2_deriv (const ex &y, const ex &x, unsigned deriv_param)
- static bool atan2_info (const ex &y, const ex &x, unsigned inf)
- REGISTER_FUNCTION (atan2, eval_func(atan2_eval). evalf_func(atan2_evalf). info_func(atan2_info). evalf_func(atan2_evalf). derivative_func(atan2_deriv))
- static ex sinh_evalf (const ex &x)
- static ex sinh_eval (const ex &x)
- static ex sinh_deriv (const ex &x, unsigned deriv_param)
- static ex sinh_real_part (const ex &x)
- static ex sinh_imag_part (const ex &x)
- static ex sinh_conjugate (const ex &x)
- REGISTER_FUNCTION (sinh, eval_func(sinh_eval). evalf_func(sinh_evalf). info_func(atan_info). derivative_func(sinh_deriv). real_part_func(sinh_real_part). imag_part_func(sinh_imag_part). conjugate_func(sinh_conjugate). latex_name("\\sinh"))
- static ex cosh_evalf (const ex &x)
- static ex cosh_eval (const ex &x)
- static ex cosh_deriv (const ex &x, unsigned deriv_param)
- static ex cosh_real_part (const ex &x)
- static ex cosh_imag_part (const ex &x)
- static ex cosh_conjugate (const ex &x)
- REGISTER_FUNCTION (cosh, eval_func(cosh_eval). evalf_func(cosh_evalf). info_func(exp_info). derivative_func(cosh_deriv). real_part_func(cosh_real_part). imag_part_func(cosh_imag_part). conjugate_func(cosh_conjugate). latex_name("\\cosh"))
- static ex tanh_evalf (const ex &x)
- static ex tanh_eval (const ex &x)
- static ex tanh_deriv (const ex &x, unsigned deriv_param)
- static ex tanh_series (const ex &x, const relational &rel, int order, unsigned options)
- static ex tanh_real_part (const ex &x)
- static ex tanh_imag_part (const ex &x)
- static ex tanh_conjugate (const ex &x)
- REGISTER_FUNCTION (tanh, eval_func(tanh_eval). evalf_func(tanh_evalf). info_func(atan_info). derivative_func(tanh_deriv). series_func(tanh_series). real_part_func(tanh_real_part). imag_part_func(tanh_imag_part). conjugate_func(tanh_conjugate). latex_name("\\tanh"))
- static ex asinh_evalf (const ex &x)
- static ex asinh_eval (const ex &x)
- static ex asinh_deriv (const ex &x, unsigned deriv_param)
- static ex asinh_conjugate (const ex &x)
- REGISTER_FUNCTION (asinh, eval_func(asinh_eval). evalf_func(asinh_evalf). info_func(atan_info). derivative_func(asinh_deriv). conjugate_func(asinh_conjugate))
- static ex acosh_evalf (const ex &x)
- static ex acosh_eval (const ex &x)
- static ex acosh_deriv (const ex &x, unsigned deriv_param)
- static ex acosh_conjugate (const ex &x)
- REGISTER_FUNCTION (acosh, eval_func(acosh_eval). evalf_func(acosh_evalf). info_func(asin_info). derivative_func(acosh_deriv). conjugate_func(acosh_conjugate))
- static ex atanh_evalf (const ex &x)
- static ex atanh_eval (const ex &x)
- static ex atanh_deriv (const ex &x, unsigned deriv_param)
- static ex atanh_series (const ex &arg, const relational &rel, int order, unsigned options)

- static ex atanh_conjugate (const ex &x)
- REGISTER_FUNCTION (atanh, eval_func(atanh_eval). evalf_func(atanh_evalf). info_func(asin_info). derivative_func(atanh_deriv). series_func(atanh_series). conjugate_func(atanh_conjugate))
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (integral, basic, print_func< print_dflt >(&integral::do_print). print_func< print_python >(&integral::do_print). print_func< print_latex >(&integral::do_print_latex)) integral
- ex subsvalue (const ex &var, const ex &value, const ex &fun)
- ex adaptivesimpson (const ex &x, const ex &a_in, const ex &b_in, const ex &f, const ex &error)

  Numeric integration routine based upon the "Adaptive Quadrature" one in "Numerical Analysis" by Burden and Faires.
- GINAC_BIND_UNARCHIVER (integral)
- GINAC_DECLARE_UNARCHIVER (integral)
- ex ifactor (const numeric &n)

  Returns the decomposition of the positive integer n into prime numbers in the form lst(lst(p1,...,pr), lst(a1,...,ar)) such that n = p1^a1 ... pr^ar.
- bool is_discriminant_of_quadratic_number_field (const numeric &n)

  Returns true if the integer n is either one or the discriminant of a quadratic number field.
- numeric kronecker_symbol (const numeric &a, const numeric &n)

  Returns the Kronecker symbol a: integer n: integer.
- numeric primitive_dirichlet_character (const numeric &n, const numeric &a)

  Defines a primitive Dirichlet character through the Kronecker symbol.
- numeric dirichlet_character (const numeric &n, const numeric &a, const numeric &N)

  Defines a Dirichlet character through the Kronecker symbol.
- numeric generalised_Bernoulli_number (const numeric &k, const numeric &b)

  The generalised Bernoulli number.
- ex Bernoulli_polynomial (const numeric &k, const ex &x)

  The Bernoulli polynomials.
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (integration_kernel, basic, print_func< print_context >(&integration_kernel::do_print)) integration_kernel
- GINAC_BIND_UNARCHIVER (integration_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (basic_log_kernel, integration_kernel, print_func< print_context >(&basic_log_kernel::do_print)) basic_log_kernel
- GINAC_BIND_UNARCHIVER (basic_log_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (multiple_polylog_kernel, integration_kernel, print_↩ func< print_context >(&multiple_polylog_kernel::do_print)) multiple_polylog_kernel
- GINAC_BIND_UNARCHIVER (multiple_polylog_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (ELi_kernel, integration_kernel, print_func< print_context >(&ELi_kernel::do_print)) ELi_kernel
- GINAC_BIND_UNARCHIVER (ELi_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Ebar_kernel, integration_kernel, print_func< print_context >(&Ebar_kernel::do_print)) Ebar_kernel
- GINAC_BIND_UNARCHIVER (Ebar_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Kronecker_dtau_kernel, integration_kernel, print_↩ func< print_context >(&Kronecker_dtau_kernel::do_print)) Kronecker_dtau_kernel
- GINAC_BIND_UNARCHIVER (Kronecker_dtau_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Kronecker_dz_kernel, integration_kernel, print_func< print_context >(&Kronecker_dz_kernel::do_print)) Kronecker_dz_kernel
- GINAC_BIND_UNARCHIVER (Kronecker_dz_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Eisenstein_kernel, integration_kernel, print_func< print_context >(&Eisenstein_kernel::do_print)) Eisenstein_kernel
- GINAC_BIND_UNARCHIVER (Eisenstein_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Eisenstein_h_kernel, integration_kernel, print_func< print_context >(&Eisenstein_h_kernel::do_print)) Eisenstein_h_kernel
- GINAC_BIND_UNARCHIVER (Eisenstein_h_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (modular_form_kernel, integration_kernel, print_func< print_context >(&modular_form_kernel::do_print)) modular_form_kernel

- GINAC_BIND_UNARCHIVER (modular_form_kernel)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (user_defined_kernel, integration_kernel, print_func< print_context >(&user_defined_kernel::do_print)) user_defined_kernel
- GINAC_BIND_UNARCHIVER (user_defined_kernel)
- GINAC_DECLARE_UNARCHIVER (integration_kernel)
- GINAC_DECLARE_UNARCHIVER (basic_log_kernel)
- GINAC_DECLARE_UNARCHIVER (multiple_polylog_kernel)
- GINAC_DECLARE_UNARCHIVER (ELi_kernel)
- GINAC_DECLARE_UNARCHIVER (Ebar_kernel)
- GINAC_DECLARE_UNARCHIVER (Kronecker_dtau_kernel)
- GINAC_DECLARE_UNARCHIVER (Kronecker_dz_kernel)
- GINAC_DECLARE_UNARCHIVER (Eisenstein_kernel)
- GINAC_DECLARE_UNARCHIVER (Eisenstein_h_kernel)
- GINAC_DECLARE_UNARCHIVER (modular_form_kernel)
- GINAC_DECLARE_UNARCHIVER (user_defined_kernel)
- GINAC_DECLARE_UNARCHIVER (lst)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (matrix, basic, print_func< print_context >(&matrix::do_print). print_func< print_latex >(&matrix::do_print_latex). print_func< print_tree >(&matrix::do_print_tree). print↩ _func< print_python_repr >(&matrix::do_print_python_repr)) matrix

  Default ctor.
- GINAC_BIND_UNARCHIVER (matrix)
- ex lst_to_matrix (const lst &l)

  Convert list of lists to matrix.
- ex diag_matrix (const lst &l)

  Convert list of diagonal elements to matrix.
- ex diag_matrix (std::initializer_list< ex > l)
- ex unit_matrix (unsigned r, unsigned c)

  Create an r times c unit matrix.
- ex symbolic_matrix (unsigned r, unsigned c, const std::string &base_name, const std::string &tex_base_↩ name)

  Create an r times c matrix of newly generated symbols consisting of the given base name plus the numeric row/column position of each element.
- ex reduced_matrix (const matrix &m, unsigned r, unsigned c)

  Return the reduced matrix that is formed by deleting the rth row and cth column of matrix m.
- ex sub_matrix (const matrix &m, unsigned r, unsigned nr, unsigned c, unsigned nc)

  Return the nr times nc submatrix starting at position r, c of matrix m.
- GINAC_DECLARE_UNARCHIVER (matrix)
- size_t nops (const matrix &m)
- ex expand (const matrix &m, unsigned options=0)
- ex evalf (const matrix &m)
- unsigned rows (const matrix &m)
- unsigned cols (const matrix &m)
- matrix transpose (const matrix &m)
- ex determinant (const matrix &m, unsigned options=determinant_algo::automatic)
- ex trace (const matrix &m)
- ex charpoly (const matrix &m, const ex &lambda)
- matrix inverse (const matrix &m)
- matrix inverse (const matrix &m, unsigned algo)
- unsigned rank (const matrix &m)
- unsigned rank (const matrix &m, unsigned solve_algo)
- ex unit_matrix (unsigned x)

  Create a x times x unit matrix.
- ex symbolic_matrix (unsigned r, unsigned c, const std::string &base_name)

Create an r times c matrix of newly generated symbols consisting of the given base name plus the numeric row/column position of each element.

- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (mul, expairseq, print_func< print_context >(&mul::do_print). print_func< print_latex >(&mul::do_print_latex). print_func< print_csrc >(&mul::do_print_csrc). print_↩ func< print_tree >(&mul::do_print_tree). print_func< print_python_repr >(&mul::do_print_python_repr)) mul
- bool tryfactsubs (const ex &origfactor, const ex &patternfactor, int &nummatches, exmap &repls)
- bool algebraic_match_mul_with_mul (const mul &e, const ex &pat, exmap &repls, int factor, int &nummatches, const std::vector< bool > &subsed, std::vector< bool > &matched)

  Checks whether e matches to the pattern pat and the (possibly to be updated) list of replacements repls.
- GINAC_BIND_UNARCHIVER (mul)
- GINAC_DECLARE_UNARCHIVER (mul)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (ncmul, exprseq, print_func< print_context >(&ncmul::do_print). print_func< print_tree >(&ncmul::do_print_tree). print_func< print_csrc >(&ncmul::do_print_csrc). print_↩ func< print_python_repr >(&ncmul::do_print_csrc)) ncmul
- ex reeval_ncmul (const exvector &v)
- ex hold_ncmul (const exvector &v)
- GINAC_BIND_UNARCHIVER (ncmul)
- GINAC_DECLARE_UNARCHIVER (ncmul)
- static bool get_first_symbol (const ex &e, ex &x)

  Return pointer to first symbol found in expression.
- static void add_symbol (const ex &s, sym_desc_vec &v)
- static void collect_symbols (const ex &e, sym_desc_vec &v)
- static void get_symbol_stats (const ex &a, const ex &b, sym_desc_vec &v)

  Collect statistical information about symbols in polynomials.
- static numeric lcmcoeff (const ex &e, const numeric &l)
- static numeric lcm_of_coefficients_denominators (const ex &e)

  Compute LCM of denominators of coefficients of a polynomial.
- static ex multiply_lcm (const ex &e, const numeric &lcm)

  Bring polynomial from Q[X] to Z[X] by multiplying in the previously determined LCM of the coefficient's denominators.
- ex quo (const ex &a, const ex &b, const ex &x, bool check_args)

  Quotient q(x) of polynomials a(x) and b(x) in Q[x].
- ex rem (const ex &a, const ex &b, const ex &x, bool check_args)

  Remainder r(x) of polynomials a(x) and b(x) in Q[x].
- ex decomp_rational (const ex &a, const ex &x)

  Decompose rational function a(x)=N(x)/D(x) into P(x)+n(x)/D(x) with degree(n, x) < degree(D, x).
- ex prem (const ex &a, const ex &b, const ex &x, bool check_args)

  Pseudo-remainder of polynomials a(x) and b(x) in Q[x].
- ex sprem (const ex &a, const ex &b, const ex &x, bool check_args)

  Sparse pseudo-remainder of polynomials a(x) and b(x) in Q[x].
- bool divide (const ex &a, const ex &b, ex &q, bool check_args)

  Exact polynomial division of a(X) by b(X) in Q[X].
- static bool divide_in_z (const ex &a, const ex &b, ex &q, sym_desc_vec::const_iterator var)

  Exact polynomial division of a(X) by b(X) in Z[X].
- static ex sr_gcd (const ex &a, const ex &b, sym_desc_vec::const_iterator var)

  Compute GCD of multivariate polynomials using the subresultant PRS algorithm.
- static ex interpolate (const ex &gamma, const numeric &xi, const ex &x, int degree_hint=1)

  xi-adic polynomial interpolation
- static bool heur_gcd_z (ex &res, const ex &a, const ex &b, ex ∗ca, ex ∗cb, sym_desc_vec::const_iterator var)

  Compute GCD of multivariate polynomials using the heuristic GCD algorithm.
- static bool heur_gcd (ex &res, const ex &a, const ex &b, ex ∗ca, ex ∗cb, sym_desc_vec::const_iterator var)

  Compute GCD of multivariate polynomials using the heuristic GCD algorithm.
- static ex gcd_pf_pow (const ex &a, const ex &b, ex ∗ca, ex ∗cb)

- static ex gcd_pf_mul (const ex &a, const ex &b, ex *ca, ex *cb)
- ex gcd (const ex &a, const ex &b, ex *ca, ex *cb, bool check_args, unsigned options)

  Compute GCD (Greatest Common Divisor) of multivariate polynomials a(X) and b(X) in Z[X].
- static ex gcd_pf_pow_pow (const ex &a, const ex &b, ex *ca, ex *cb)
- ex lcm (const ex &a, const ex &b, bool check_args)

  Compute LCM (Least Common Multiple) of multivariate polynomials in Z[X].
- static epvector sqrfree_yun (const ex &a, const symbol &x)

  Compute square-free factorization of multivariate polynomial a(x) using Yun's algorithm.
- ex sqrfree (const ex &a, const lst &l)

  Compute a square-free factorization of a multivariate polynomial in Q[X].
- ex sqrfree_parfrac (const ex &a, const symbol &x)

  Compute square-free partial fraction decomposition of rational function a(x).
- static ex replace_with_symbol (const ex &e, exmap &repl, exmap &rev_lookup, lst &modifier)

  Create a symbol for replacing the expression "e" (or return a previously assigned symbol).
- static ex replace_with_symbol (const ex &e, exmap &repl)

  Create a symbol for replacing the expression "e" (or return a previously assigned symbol).
- static ex frac_cancel (const ex &n, const ex &d)

  Fraction cancellation.
- static ex find_common_factor (const ex &e, ex &factor, exmap &repl)

  Remove the common factor in the terms of a sum 'e' by calculating the GCD, and multiply it into the expression 'factor' (which needs to be initialized to 1, unless you're accumulating factors).
- ex collect_common_factors (const ex &e)

  Collect common factors in sums.
- ex resultant (const ex &e1, const ex &e2, const ex &s)

  Resultant of two expressions e1,e2 with respect to symbol s.
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (numeric, basic, print_func< print_context >(&numeric::do_print). print_func< print_latex >(&numeric::do_print_latex). print_func< print_csrc >(&numeric::do_print_csrc). print_func< print_csrc_cl_N >(&numeric::do_print_csrc_cl_N). print_func< print_tree >(&numeric::do_print_tree). print_func< print_python_repr >(&numeric::do_print_python_repr)) numeric

  default ctor.
- static const cln::cl_F make_real_float (const cln::cl_idecoded_float &dec)

  Construct a floating point number from sign, mantissa, and exponent.
- static const cln::cl_F read_real_float (std::istream &s)

  Read serialized floating point number.
- GINAC_BIND_UNARCHIVER (numeric)
- static void write_real_float (std::ostream &s, const cln::cl_R &n)
- static void print_real_number (const print_context &c, const cln::cl_R &x)

  Helper function to print a real number in a nicer way than is CLN's default.
- static void print_integer_csrc (const print_context &c, const cln::cl_I &x)

  Helper function to print integer number in C++ source format.
- static void print_real_csrc (const print_context &c, const cln::cl_R &x)

  Helper function to print real number in C++ source format.
- template<typename T1, typename T2>
  static bool coerce (T1 &dst, const T2 &arg)
- template<> bool coerce< int, cln::cl_I > (int &dst, const cln::cl_I &arg)

  Check if CLN integer can be converted into int.
- template<> bool coerce< unsigned int, cln::cl_I > (unsigned int &dst, const cln::cl_I &arg)
- static void print_real_cl_N (const print_context &c, const cln::cl_R &x)

  Helper function to print real number in C++ source format using cl_N types.
- const numeric exp (const numeric &x)

  Exponential function.
- const numeric log (const numeric &x)

Natural logarithm.

- const numeric sin (const numeric &x)

  Numeric sine (trigonometric function).
- const numeric cos (const numeric &x)

  Numeric cosine (trigonometric function).
- const numeric tan (const numeric &x)

  Numeric tangent (trigonometric function).
- const numeric asin (const numeric &x)

  Numeric inverse sine (trigonometric function).
- const numeric acos (const numeric &x)

  Numeric inverse cosine (trigonometric function).
- const numeric atan (const numeric &x)

  Numeric arcustangent.
- const numeric atan (const numeric &y, const numeric &x)

  Numeric arcustangent of two arguments, analytically continued in a suitable way.
- const numeric sinh (const numeric &x)

  Numeric hyperbolic sine (trigonometric function).
- const numeric cosh (const numeric &x)

  Numeric hyperbolic cosine (trigonometric function).
- const numeric tanh (const numeric &x)

  Numeric hyperbolic tangent (trigonometric function).
- const numeric asinh (const numeric &x)

  Numeric inverse hyperbolic sine (trigonometric function).
- const numeric acosh (const numeric &x)

  Numeric inverse hyperbolic cosine (trigonometric function).
- const numeric atanh (const numeric &x)

  Numeric inverse hyperbolic tangent (trigonometric function).
- static cln::cl_N Li2_series (const cln::cl_N &x, const cln::float_format_t &prec)

  Numeric evaluation of Dilogarithm within circle of convergence (unit circle) using a power series.
- static cln::cl_N Li2_projection (const cln::cl_N &x, const cln::float_format_t &prec)

  Folds Li2's argument inside a small rectangle to enhance convergence.
- const cln::cl_N Li2_ (const cln::cl_N &value)

  Numeric evaluation of Dilogarithm.
- const numeric Li2 (const numeric &x)
- const numeric zeta (const numeric &x)

  Numeric evaluation of Riemann's Zeta function.
- static cln::float_format_t guess_precision (const cln::cl_N &x)
- const cln::cl_N lgamma (const cln::cl_N &x)

  The Gamma function.
- const numeric lgamma (const numeric &x)
- const cln::cl_N tgamma (const cln::cl_N &x)
- const numeric tgamma (const numeric &x)
- const numeric psi (const numeric &x)

  The psi function (aka polygamma function).
- const numeric psi (const numeric &n, const numeric &x)

  The psi functions (aka polygamma functions).
- const numeric factorial (const numeric &n)

  Factorial combinatorial function.
- const numeric doublefactorial (const numeric &n)

  The double factorial combinatorial function.
- const numeric binomial (const numeric &n, const numeric &k)

The Binomial coefficients.

- const numeric bernoulli (const numeric &nn)

  Bernoulli number.
- const numeric fibonacci (const numeric &n)

  Fibonacci number.
- const numeric abs (const numeric &x)

  Absolute value.
- const numeric mod (const numeric &a, const numeric &b)

  Modulus (in positive representation).
- const numeric smod (const numeric &a_, const numeric &b_)

  Modulus (in symmetric representation).
- const numeric irem (const numeric &a, const numeric &b)

  Numeric integer remainder.
- const numeric irem (const numeric &a, const numeric &b, numeric &q)

  Numeric integer remainder.
- const numeric iquo (const numeric &a, const numeric &b)

  Numeric integer quotient.
- const numeric iquo (const numeric &a, const numeric &b, numeric &r)

  Numeric integer quotient.
- const numeric gcd (const numeric &a, const numeric &b)

  Greatest Common Divisor.
- const numeric lcm (const numeric &a, const numeric &b)

  Least Common Multiple.
- const numeric sqrt (const numeric &x)

  Numeric square root.
- const numeric isqrt (const numeric &x)

  Integer numeric square root.
- ex PiEvalf ()

  Floating point evaluation of Archimedes' constant Pi.
- ex EulerEvalf ()

  Floating point evaluation of Euler's constant gamma.
- ex CatalanEvalf ()

  Floating point evaluation of Catalan's constant.
- std::ostream & operator<< (std::ostream &os, const _numeric_digits &e)
- GINAC_DECLARE_UNARCHIVER (numeric)
- const numeric pow (const numeric &x, const numeric &y)
- const numeric inverse (const numeric &x)
- numeric step (const numeric &x)
- int csgn (const numeric &x)
- bool is_zero (const numeric &x)
- bool is_positive (const numeric &x)
- bool is_negative (const numeric &x)
- bool is_integer (const numeric &x)
- bool is_pos_integer (const numeric &x)
- bool is_nonneg_integer (const numeric &x)
- bool is_even (const numeric &x)
- bool is_odd (const numeric &x)
- bool is_prime (const numeric &x)
- bool is_rational (const numeric &x)
- bool is_real (const numeric &x)
- bool is_cinteger (const numeric &x)
- bool is_crational (const numeric &x)

- int to_int (const numeric &x)
- long to_long (const numeric &x)
- double to_double (const numeric &x)
- const numeric real (const numeric &x)
- const numeric imag (const numeric &x)
- const numeric numer (const numeric &x)
- const numeric denom (const numeric &x)
- static const ex exadd (const ex &lh, const ex &rh)

  Used internally by operator+() to add two ex objects.
- static const ex exmul (const ex &lh, const ex &rh)

  Used internally by operator∗() to multiply two ex objects.
- static const ex exminus (const ex &lh)

  Used internally by operator-() and friends to change the sign of an argument.
- const ex operator+ (const ex &lh, const ex &rh)
- const ex operator- (const ex &lh, const ex &rh)
- const ex operator∗ (const ex &lh, const ex &rh)
- const ex operator/ (const ex &lh, const ex &rh)
- const numeric operator+ (const numeric &lh, const numeric &rh)
- const numeric operator- (const numeric &lh, const numeric &rh)
- const numeric operator∗ (const numeric &lh, const numeric &rh)
- const numeric operator/ (const numeric &lh, const numeric &rh)
- ex & operator+= (ex &lh, const ex &rh)
- ex & operator-= (ex &lh, const ex &rh)
- ex & operator∗= (ex &lh, const ex &rh)
- ex & operator/= (ex &lh, const ex &rh)
- numeric & operator+= (numeric &lh, const numeric &rh)
- numeric & operator-= (numeric &lh, const numeric &rh)
- numeric & operator∗= (numeric &lh, const numeric &rh)
- numeric & operator/= (numeric &lh, const numeric &rh)
- const ex operator+ (const ex &lh)
- const ex operator- (const ex &lh)
- const numeric operator+ (const numeric &lh)
- const numeric operator- (const numeric &lh)
- ex & operator++ (ex &rh)

  Expression prefix increment.
- ex & operator-- (ex &rh)

  Expression prefix decrement.
- const ex operator++ (ex &lh, int)

  Expression postfix increment.
- const ex operator-- (ex &lh, int)

  Expression postfix decrement.
- numeric & operator++ (numeric &rh)

  Numeric prefix increment.
- numeric & operator-- (numeric &rh)

  Numeric prefix decrement.
- const numeric operator++ (numeric &lh, int)

  Numeric postfix increment.
- const numeric operator-- (numeric &lh, int)

  Numeric postfix decrement.
- const relational operator== (const ex &lh, const ex &rh)
- const relational operator!= (const ex &lh, const ex &rh)
- const relational operator< (const ex &lh, const ex &rh)
- const relational operator<= (const ex &lh, const ex &rh)

- const relational operator> (const ex &lh, const ex &rh)
- const relational operator>= (const ex &lh, const ex &rh)
- static int my_ios_index ()
- static void my_ios_callback (std::ios_base::event ev, std::ios_base &s, int i)
- static print_context * get_print_context (std::ios_base &s)
- static void set_print_context (std::ios_base &s, const print_context &c)
- static unsigned get_print_options (std::ios_base &s)
- static void set_print_options (std::ostream &s, unsigned options)
- std::ostream & operator<< (std::ostream &os, const ex &e)
- std::istream & operator>> (std::istream &is, ex &e)
- std::ostream & dflt (std::ostream &os)
- std::ostream & latex (std::ostream &os)
- std::ostream & python (std::ostream &os)
- std::ostream & python_repr (std::ostream &os)
- std::ostream & tree (std::ostream &os)
- std::ostream & csrc (std::ostream &os)
- std::ostream & csrc_float (std::ostream &os)
- std::ostream & csrc_double (std::ostream &os)
- std::ostream & csrc_cl_N (std::ostream &os)
- std::ostream & index_dimensions (std::ostream &os)
- std::ostream & no_index_dimensions (std::ostream &os)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (power, basic, print_func< print_dflt >(&power::do_print_dflt). print_func< print_latex >(&power::do_print_latex). print_func< print_csrc >(&power::do_print_csrc). print↩ _func< print_python >(&power::do_print_python). print_func< print_python_repr >(&power::do_print_python_repr). print_func< print_csrc_cl_N >(&power::do_print_csrc_cl_N)) power
- static void print_sym_pow (const print_context &c, const symbol &x, int exp)
- GINAC_BIND_UNARCHIVER (power)
- GINAC_DECLARE_UNARCHIVER (power)
- ex pow (const ex &b, const ex &e)

  Symbolic exponentiation.
- template<typename T1, typename T2>
  ex pow (const T1 &b, const T2 &e)
- ex sqrt (const ex &a)

  Square root expression.
- template<class T>
  bool is_a (const print_context &obj)

  Check if obj is a T, including base classes.
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (pseries, basic, print_func< print_context >(&pseries::do_print). print_func< print_latex >(&pseries::do_print_latex). print_func< print_tree >(&pseries::do_print_tree). print_func< print_python >(&pseries::do_print_python). print_func< print_python_repr >(&pseries::do_print_python_repr)) pseries
- GINAC_BIND_UNARCHIVER (pseries)
- GINAC_DECLARE_UNARCHIVER (pseries)
- ex series_to_poly (const ex &e)

  Convert the pseries object embedded in an expression to an ordinary polynomial in the expansion variable.
- bool is_terminating (const pseries &s)
- template<typename T>
  return_type_t make_return_type_t (const unsigned rl=0)
- template<class Alg, class Ctx, class T, class C>
  void set_print_func (void f(const T &, const C &c, unsigned))

  Add or replace a print method.
- template<class Alg, class Ctx, class T, class C>
  void set_print_func (void(T::*f)(const C &, unsigned))

  Add or replace a print method.

- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (relational, basic, print_func< print_context >(&relational::do_print). print_func< print_tree >(&relational::do_print_tree). print_func< print_python_repr >(&relational::do_print_python_repr)) relational
- GINAC_BIND_UNARCHIVER (relational)
- static void print_operator (const print_context &c, relational::operators o)
- GINAC_DECLARE_UNARCHIVER (relational)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (symbol, basic, print_func< print_context >(&symbol::do_print). print_func< print_latex >(&symbol::do_print_latex). print_func< print_tree >(&symbol::do_print_tree). print_func< print_python_repr >(&symbol::do_print_python_repr)) symbol
- static const std::string & get_default_TeX_name (const std::string &name)

  Return default TeX name for symbol.
- GINAC_BIND_UNARCHIVER (symbol)
- GINAC_BIND_UNARCHIVER (realsymbol)
- GINAC_BIND_UNARCHIVER (possymbol)
- GINAC_DECLARE_UNARCHIVER (symbol)
- GINAC_DECLARE_UNARCHIVER (realsymbol)
- GINAC_DECLARE_UNARCHIVER (possymbol)
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (symmetry, basic, print_func< print_context >(&symmetry::do_print). print_func< print_tree >(&symmetry::do_print_tree)) symmetry
- GINAC_BIND_UNARCHIVER (symmetry)
- static const symmetry & index0 ()
- static const symmetry & index1 ()
- static const symmetry & index2 ()
- static const symmetry & index3 ()
- const symmetry & not_symmetric ()
- const symmetry & symmetric2 ()
- const symmetry & symmetric3 ()
- const symmetry & symmetric4 ()
- const symmetry & antisymmetric2 ()
- const symmetry & antisymmetric3 ()
- const symmetry & antisymmetric4 ()
- int canonicalize (exvector::iterator v, const symmetry &symm)

  Canonicalize the order of elements of an expression vector, according to the symmetry properties defined in a symmetry tree.
- static ex symm (const ex &e, exvector::const_iterator first, exvector::const_iterator last, bool asymmetric)
- ex symmetrize (const ex &e, exvector::const_iterator first, exvector::const_iterator last)

  Symmetrize expression over a set of objects (symbols, indices).
- ex antisymmetrize (const ex &e, exvector::const_iterator first, exvector::const_iterator last)

  Antisymmetrize expression over a set of objects (symbols, indices).
- ex symmetrize_cyclic (const ex &e, exvector::const_iterator first, exvector::const_iterator last)

  Symmetrize expression by cyclic permutation over a set of objects (symbols, indices).
- GINAC_DECLARE_UNARCHIVER (symmetry)
- symmetry sy_none ()
- symmetry sy_none (const symmetry &c1, const symmetry &c2)
- symmetry sy_none (const symmetry &c1, const symmetry &c2, const symmetry &c3)
- symmetry sy_none (const symmetry &c1, const symmetry &c2, const symmetry &c3, const symmetry &c4)
- symmetry sy_symm ()
- symmetry sy_symm (const symmetry &c1, const symmetry &c2)
- symmetry sy_symm (const symmetry &c1, const symmetry &c2, const symmetry &c3)
- symmetry sy_symm (const symmetry &c1, const symmetry &c2, const symmetry &c3, const symmetry &c4)
- symmetry sy_anti ()
- symmetry sy_anti (const symmetry &c1, const symmetry &c2)
- symmetry sy_anti (const symmetry &c1, const symmetry &c2, const symmetry &c3)
- symmetry sy_anti (const symmetry &c1, const symmetry &c2, const symmetry &c3, const symmetry &c4)

- symmetry sy_cycl ()
- symmetry sy_cycl (const symmetry &c1, const symmetry &c2)
- symmetry sy_cycl (const symmetry &c1, const symmetry &c2, const symmetry &c3)
- symmetry sy_cycl (const symmetry &c1, const symmetry &c2, const symmetry &c3, const symmetry &c4)
- ex symmetrize (const ex &e, const exvector &v)

  Symmetrize expression over a set of objects (symbols, indices).
- ex antisymmetrize (const ex &e, const exvector &v)

  Antisymmetrize expression over a set of objects (symbols, indices).
- ex symmetrize_cyclic (const ex &e, const exvector &v)

  Symmetrize expression by cyclic permutation over a set of objects (symbols, indices).
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (tensdelta, tensor, print_func< print_dflt >(&tensdelta::do_print). print_func< print_latex >(&tensdelta::do_print_latex)) GINAC_IMPLEMENT_REGISTERED_CLASS_↩ OPT(tensmetric
- print_func< print_dflt > (&tensmetric::do_print). print_func< print_latex >(&tensmetric
- GINAC_BIND_UNARCHIVER (minkmetric)
- GINAC_BIND_UNARCHIVER (tensepsilon)
- GINAC_BIND_UNARCHIVER (tensdelta)
- GINAC_BIND_UNARCHIVER (tensmetric)
- GINAC_BIND_UNARCHIVER (spinmetric)
- ex delta_tensor (const ex &i1, const ex &i2)

  Create a delta tensor with specified indices.
- ex metric_tensor (const ex &i1, const ex &i2)

  Create a symmetric metric tensor with specified indices.
- ex lorentz_g (const ex &i1, const ex &i2, bool pos_sig=false)

  Create a Minkowski metric tensor with specified indices.
- ex spinor_metric (const ex &i1, const ex &i2)

  Create a spinor metric tensor with specified indices.
- ex epsilon_tensor (const ex &i1, const ex &i2)

  Create an epsilon tensor in a Euclidean space with two indices.
- ex epsilon_tensor (const ex &i1, const ex &i2, const ex &i3)

  Create an epsilon tensor in a Euclidean space with three indices.
- ex lorentz_eps (const ex &i1, const ex &i2, const ex &i3, const ex &i4, bool pos_sig=false)

  Create an epsilon tensor in a Minkowski space with four indices.
- GINAC_DECLARE_UNARCHIVER (tensdelta)
- GINAC_DECLARE_UNARCHIVER (tensmetric)
- GINAC_DECLARE_UNARCHIVER (minkmetric)
- GINAC_DECLARE_UNARCHIVER (spinmetric)
- GINAC_DECLARE_UNARCHIVER (tensepsilon)
- unsigned log2 (unsigned n)

  Integer binary logarithm.
- const numeric multinomial_coefficient (const std::vector< unsigned > &p)

  Compute the multinomial coefficient n!
- unsigned rotate_left (unsigned n)

  Rotate bits of unsigned value by one bit to the left.
- template<class T>
  int compare_pointers (const T *a, const T *b)

  Compare two pointers (just to establish some sort of canonical order).
- unsigned golden_ratio_hash (uintptr_t n)

  Truncated multiplication with golden ratio, for computing hash values.
- template<class It>
  int permutation_sign (It first, It last)
- template<class It, class Cmp, class Swap>
  int permutation_sign (It first, It last, Cmp comp, Swap swapit)

- template<class It, class Cmp, class Swap>
  void shaker_sort (It first, It last, Cmp comp, Swap swapit)
- template<class It, class Swap>
  void cyclic_permutation (It first, It last, It new_first, Swap swapit)
- template<typename T>
  std::enable_if< has_distance< T >::value, typenamestd::iterator_traits< T >::difference_type >::type format_index_value (const T &a, const T &b)

  For printing a multi-index: If the templates are used, where T is an iterator, printing the address where the iterator points to is not meaningful.
- template<typename T>
  std::enable_if<!has_distance< T >::value, T >::type format_index_value (const T &a, const T &b)

  For all other cases we simply print the value.
- template<class T>
  std::ostream & operator<< (std::ostream &os, const basic_multi_iterator< T > &v)

  Output operator.
- template<class T>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_ordered< T > &v)

  Output operator.
- template<class T>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_ordered_eq< T > &v)

  Output operator.
- template<class T>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_ordered_eq_indv< T > &v)

  Output operator.
- template<class T>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_counter< T > &v)

  Output operator.
- template<class T>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_counter_indv< T > &v)

  Output operator.
- template<class T>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_permutation< T > &v)

  Output operator.
- template<class T>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_shuffle< T > &v)

  Output operator.
- template<class T>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_shuffle_prime< T > &v)

  Output operator.
- GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (wildcard, basic, print_func< print_context >(&wildcard::do_print). print_func< print_tree >(&wildcard::do_print_tree). print_func< print_python_repr >(&wildcard::do_print_python_repr))
  wildcard
- GINAC_BIND_UNARCHIVER (wildcard)
- bool haswild (const ex &x)

  Check whether x has a wildcard anywhere as a subexpression.
- GINAC_DECLARE_UNARCHIVER (wildcard)
- ex wild (unsigned label=0)

  Create a wildcard object with the specified label.

Variables

- static unarchive_table_t unarch_table_instance
- GiNaC::evalm_map_function map_evalm
- GiNaC::eval_integ_map_function map_eval_integ
- template<class OPT>
  class_info< OPT > * class_info< OPT >::first = nullptr
- template<class OPT>
  bool class_info< OPT >::parents_identified = false
- tensor
- const constant Pi ("Pi", PiEvalf, "\\pi", domain::positive)

  Pi.
- const constant Euler ("Euler", EulerEvalf, "\\gamma_E", domain::positive)

  Euler's constant.
- const constant Catalan ("Catalan", CatalanEvalf, "G", domain::positive)

  Catalan's constant.
- static unsigned const crctab [256]
- static library_init library_initializer

  For construction of flyweights, etc.
- const basic * _num0_bp
- idx
- unsigned force_include_tgamma = tgamma_SERIAL::serial
- unsigned force_include_zeta1 = zeta1_SERIAL::serial
- template<> GINAC_IMPLEMENT_REGISTERED_CLASS_OPT_T(lst, basic, print_func< print_context >(&lst::do_print). print_func< print_tree >(&lst::do_print_tree)) template<> bool lst GINAC_BIND_UNARCHIVER (lst)

  Specialization of container::info() for lst.
- const numeric I = numeric(cln::complex(cln::cl_I(0),cln::cl_I(1)))

  Imaginary unit.
- _numeric_digits Digits

  Accuracy in decimal digits.
- unsigned next_print_context_id = 0

  Next free ID for print_context types.
- template<class T, template< class > class CP>
  registered_class_info structure< T, CP >::reg_info = registered_class_info(registered_class_options(structure::get_class_nam "basic", typeid(structure<T, CP>)))
- const int version_major = GINACLIB_MAJOR_VERSION
- const int version_minor = GINACLIB_MINOR_VERSION
- const int version_micro = GINACLIB_MICRO_VERSION
- const numeric * _num_120_p
- const ex _ex_120 = ex(*_num_120_p)
- const numeric * _num_60_p
- const ex _ex_60 = ex(*_num_60_p)
- const numeric * _num_48_p
- const ex _ex_48 = ex(*_num_48_p)
- const numeric * _num_30_p
- const ex _ex_30 = ex(*_num_30_p)
- const numeric * _num_25_p
- const ex _ex_25 = ex(*_num_25_p)
- const numeric * _num_24_p
- const ex _ex_24 = ex(*_num_24_p)
- const numeric * _num_20_p
- const ex _ex_20 = ex(*_num_20_p)
- const numeric * _num_18_p

- const ex _ex_18 = ex(*_num_18_p)
- const numeric * _num_15_p
- const ex _ex_15 = ex(*_num_15_p)
- const numeric * _num_12_p
- const ex _ex_12 = ex(*_num_12_p)
- const numeric * _num_11_p
- const ex _ex_11 = ex(*_num_11_p)
- const numeric * _num_10_p
- const ex _ex_10 = ex(*_num_10_p)
- const numeric * _num_9_p
- const ex _ex_9 = ex(*_num_9_p)
- const numeric * _num_8_p
- const ex _ex_8 = ex(*_num_8_p)
- const numeric * _num_7_p
- const ex _ex_7 = ex(*_num_7_p)
- const numeric * _num_6_p
- const ex _ex_6 = ex(*_num_6_p)
- const numeric * _num_5_p
- const ex _ex_5 = ex(*_num_5_p)
- const numeric * _num_4_p
- const ex _ex_4 = ex(*_num_4_p)
- const numeric * _num_3_p
- const ex _ex_3 = ex(*_num_3_p)
- const numeric * _num_2_p
- const ex _ex_2 = ex(*_num_2_p)
- const numeric * _num_1_p
- const ex _ex_1 = ex(*_num_1_p)
- const numeric * _num_1_2_p
- const ex _ex_1_2 = ex(*_num_1_2_p)
- const numeric * _num_1_3_p
- const ex _ex_1_3 = ex(*_num_1_3_p)
- const numeric * _num_1_4_p
- const ex _ex_1_4 = ex(*_num_1_4_p)
- const numeric * _num0_p
- const ex _ex0 = ex(*_num0_p)
- const numeric * _num1_4_p
- const ex _ex1_4 = ex(*_num1_4_p)
- const numeric * _num1_3_p
- const ex _ex1_3 = ex(*_num1_3_p)
- const numeric * _num1_2_p
- const ex _ex1_2 = ex(*_num1_2_p)
- const numeric * _num1_p
- const ex _ex1 = ex(*_num1_p)
- const numeric * _num2_p
- const ex _ex2 = ex(*_num2_p)
- const numeric * _num3_p
- const ex _ex3 = ex(*_num3_p)
- const numeric * _num4_p
- const ex _ex4 = ex(*_num4_p)
- const numeric * _num5_p
- const ex _ex5 = ex(*_num5_p)
- const numeric * _num6_p
- const ex _ex6 = ex(*_num6_p)
- const numeric * _num7_p
- const ex _ex7 = ex(*_num7_p)

- const numeric * _num8_p
- const ex _ex8 = ex(*_num8_p)
- const numeric * _num9_p
- const ex _ex9 = ex(*_num9_p)
- const numeric * _num10_p
- const ex _ex10 = ex(*_num10_p)
- const numeric * _num11_p
- const ex _ex11 = ex(*_num11_p)
- const numeric * _num12_p
- const ex _ex12 = ex(*_num12_p)
- const numeric * _num15_p
- const ex _ex15 = ex(*_num15_p)
- const numeric * _num18_p
- const ex _ex18 = ex(*_num18_p)
- const numeric * _num20_p
- const ex _ex20 = ex(*_num20_p)
- const numeric * _num24_p
- const ex _ex24 = ex(*_num24_p)
- const numeric * _num25_p
- const ex _ex25 = ex(*_num25_p)
- const numeric * _num30_p
- const ex _ex30 = ex(*_num30_p)
- const numeric * _num48_p
- const ex _ex48 = ex(*_num48_p)
- const numeric * _num60_p
- const ex _ex60 = ex(*_num60_p)
- const numeric * _num120_p
- const ex _ex120 = ex(*_num120_p)

7.1.1 Typedef Documentation

7.1.1.1 archive_node_id

```
typedef unsigned GiNaC::archive_node_id
```

Numerical ID value to refer to an archive_node.

7.1.1.2 archive_atom

```
typedef unsigned GiNaC::archive_atom
```

Numerical ID value to refer to a string.

7.1.1.3 synthesize_func

```
typedef basic *(*) GiNaC::synthesize_func()
```

7.1.1.4 unarchive_map_t

```
typedef std::map<std::string, synthesize_func> GiNaC::unarchive_map_t
```

7.1.1.5 exvector

```
typedef std::vector<ex> GiNaC::exvector
```

7.1.1.6 exset

```
typedef std::set<ex, ex_is_less> GiNaC::exset
```

7.1.1.7 exmap

```
typedef std::map<ex, ex, ex_is_less> GiNaC::exmap
```

7.1.1.8 evalffunctype

```
typedef ex(*) GiNaC::evalffunctype()
```

7.1.1.9 FUNCP_1P

```
typedef double(*) GiNaC::FUNCP_1P(double)
```

Function pointer with one function parameter.

7.1.1.10 FUNCP_2P

```
typedef double(*) GiNaC::FUNCP_2P(double, double)
```

Function pointer with two function parameters.

7.1.1.11 FUNCP_CUBA

```
typedef void(*) GiNaC::FUNCP_CUBA(const int *, const double[], const int *, double[])
```

Function pointer for use with the CUBA library (http://www.feynarts.de/cuba).

7.1.1.12 epvector

```
typedef std::vector<expair> GiNaC::epvector
```

expair-vector

7.1.1.13 epp

```
typedef epvector::iterator GiNaC::epp
```

expair-vector pointer

7.1.1.14 exprseq

```
typedef container<std::vector> GiNaC::exprseq
```

7.1.1.15 paramset

```
typedef std::multiset<unsigned> GiNaC::paramset
```

7.1.1.16 eval_funcp

```
typedef ex(*) GiNaC::eval_funcp()
```

7.1.1.17 evalf_funcp

```
typedef ex(*) GiNaC::evalf_funcp()
```

7.1.1.18 conjugate_funcp

```
typedef ex(*) GiNaC::conjugate_funcp()
```

7.1.1.19 real_part_funcp

```
typedef ex(*) GiNaC::real_part_funcp()
```

7.1.1.20 imag_part_funcp

```
typedef ex(*) GiNaC::imag_part_funcp()
```

7.1.1.21 expand_funcp

```
typedef ex(*) GiNaC::expand_funcp()
```

7.1.1.22 derivative_funcp

```
typedef ex(*) GiNaC::derivative_funcp()
```

7.1.1.23 expl_derivative_funcp

```
typedef ex(*) GiNaC::expl_derivative_funcp()
```

7.1.1.24 power_funcp

```
typedef ex(*) GiNaC::power_funcp()
```

7.1.1.25 series_funcp

```
typedef ex(*) GiNaC::series_funcp()
```

7.1.1.26 print_funcp

```
typedef void(*) GiNaC::print_funcp()
```

7.1.1.27 info_funcp

```
typedef bool(*) GiNaC::info_funcp()
```

7.1.1.28 eval_funcp_1

```
typedef ex(*) GiNaC::eval_funcp_1(const ex &)
```

7.1.1.29 evalf_funcp_1

```
typedef ex(*) GiNaC::evalf_funcp_1(const ex &)
```

7.1.1.30 conjugate_funcp_1

```
typedef ex(*) GiNaC::conjugate_funcp_1(const ex &)
```

7.1.1.31 real_part_funcp_1

```
typedef ex(*) GiNaC::real_part_funcp_1(const ex &)
```

7.1.1.32 imag_part_funcp_1

```
typedef ex(*) GiNaC::imag_part_funcp_1(const ex &)
```

7.1.1.33 expand_funcp_1

```
typedef ex(*) GiNaC::expand_funcp_1(const ex &, unsigned)
```

7.1.1.34 derivative_funcp_1

```
typedef ex(*) GiNaC::derivative_funcp_1(const ex &, unsigned)
```

7.1.1.35 expl_derivative_funcp_1

```
typedef ex(*) GiNaC::expl_derivative_funcp_1(const ex &, const symbol &)
```

7.1.1.36 power_funcp_1

```
typedef ex(*) GiNaC::power_funcp_1(const ex &, const ex &)
```

7.1.1.37 series_funcp_1

```
typedef ex(*) GiNaC::series_funcp_1(const ex &, const relational &, int, unsigned)
```

7.1.1.38 print_funcp_1

```
typedef void(*) GiNaC::print_funcp_1(const ex &, const print_context &)
```

7.1.1.39 info_funcp_1

```
typedef bool(*) GiNaC::info_funcp_1(const ex &, unsigned)
```

7.1.1.40 eval_funcp_2

```
typedef ex(*) GiNaC::eval_funcp_2(const ex &, const ex &)
```

7.1.1.41 evalf_funcp_2

```
typedef ex(*) GiNaC::evalf_funcp_2(const ex &, const ex &)
```

7.1.1.42 conjugate_funcp_2

```
typedef ex(*) GiNaC::conjugate_funcp_2(const ex &, const ex &)
```

7.1.1.43 real_part_funcp_2

```
typedef ex(*) GiNaC::real_part_funcp_2(const ex &, const ex &)
```

7.1.1.44 imag_part_funcp_2

```
typedef ex(*) GiNaC::imag_part_funcp_2(const ex &, const ex &)
```

7.1.1.45 expand_funcp_2

```
typedef ex(*) GiNaC::expand_funcp_2(const ex &, const ex &, unsigned)
```

7.1.1.46 derivative_funcp_2

```
typedef ex(*) GiNaC::derivative_funcp_2(const ex &, const ex &, unsigned)
```

7.1.1.47 expl_derivative_funcp_2

```
typedef ex(*) GiNaC::expl_derivative_funcp_2(const ex &, const ex &, const symbol &)
```

7.1.1.48 power_funcp_2

```
typedef ex(*) GiNaC::power_funcp_2(const ex &, const ex &, const ex &)
```

7.1.1.49 series_funcp_2

```
typedef ex(*) GiNaC::series_funcp_2(const ex &, const ex &, const relational &, int, unsigned)
```

7.1.1.50 print_funcp_2

```
typedef void(*) GiNaC::print_funcp_2(const ex &, const ex &, const print_context &)
```

7.1.1.51 info_funcp_2

```
typedef bool(*) GiNaC::info_funcp_2(const ex &, const ex &, unsigned)
```

7.1.1.52 eval_funcp_3

```
typedef ex(*) GiNaC::eval_funcp_3(const ex &, const ex &, const ex &)
```

7.1.1.53 evalf_funcp_3

```
typedef ex(*) GiNaC::evalf_funcp_3(const ex &, const ex &, const ex &)
```

7.1.1.54 conjugate_funcp_3

```
typedef ex(*) GiNaC::conjugate_funcp_3(const ex &, const ex &, const ex &)
```

7.1.1.55 real_part_funcp_3

```
typedef ex(*) GiNaC::real_part_funcp_3(const ex &, const ex &, const ex &)
```

7.1.1.56 imag_part_funcp_3

```
typedef ex(*) GiNaC::imag_part_funcp_3(const ex &, const ex &, const ex &)
```

7.1.1.57 expand_funcp_3

```
typedef ex(*) GiNaC::expand_funcp_3(const ex &, const ex &, const ex &, unsigned)
```

7.1.1.58 derivative_funcp_3

```
typedef ex(*) GiNaC::derivative_funcp_3(const ex &, const ex &, const ex &, unsigned)
```

7.1.1.59 expl_derivative_funcp_3

```
typedef ex(*) GiNaC::expl_derivative_funcp_3(const ex &, const ex &, const ex &, const symbol
&)
```

7.1.1.60 power_funcp_3

```
typedef ex(*) GiNaC::power_funcp_3(const ex &, const ex &, const ex &, const ex &)
```

7.1.1.61 series_funcp_3

```
typedef ex(*) GiNaC::series_funcp_3(const ex &, const ex &, const ex &, const relational &,
int, unsigned)
```

7.1.1.62 print_funcp_3

```
typedef void(*) GiNaC::print_funcp_3(const ex &, const ex &, const ex &, const print_context
&)
```

7.1.1.63 info_funcp_3

```
typedef bool(*) GiNaC::info_funcp_3(const ex &, const ex &, const ex &, unsigned)
```

7.1.1.64 eval_funcp_4

```
typedef ex(*) GiNaC::eval_funcp_4(const ex &, const ex &, const ex &, const ex &)
```

7.1.1.65 evalf_funcp_4

```
typedef ex(*) GiNaC::evalf_funcp_4(const ex &, const ex &, const ex &, const ex &)
```

7.1.1.66 conjugate_funcp_4

```
typedef ex(*) GiNaC::conjugate_funcp_4(const ex &, const ex &, const ex &, const ex &)
```

7.1.1.67 real_part_funcp_4

```
typedef ex(*) GiNaC::real_part_funcp_4(const ex &, const ex &, const ex &, const ex &)
```

7.1.1.68 imag_part_funcp_4

```
typedef ex(*) GiNaC::imag_part_funcp_4(const ex &, const ex &, const ex &, const ex &)
```

7.1.1.69 expand_funcp_4

```
typedef ex(*) GiNaC::expand_funcp_4(const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.70 derivative_funcp_4

```
typedef ex(*) GiNaC::derivative_funcp_4(const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.71 expl_derivative_funcp_4

```
typedef ex(*) GiNaC::expl_derivative_funcp_4(const ex &, const ex &, const ex &, const ex &,
const symbol &)
```

7.1.1.72 power_funcp_4

```
typedef ex(*) GiNaC::power_funcp_4(const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.73 series_funcp_4

```
typedef ex(*) GiNaC::series_funcp_4(const ex &, const ex &, const ex &, const ex &, const
relational &, int, unsigned)
```

7.1.1.74 print_funcp_4

```
typedef void(*) GiNaC::print_funcp_4(const ex &, const ex &, const ex &, const ex &, const
print_context &)
```

7.1.1.75 info_funcp_4

```
typedef bool(*) GiNaC::info_funcp_4(const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.76 eval_funcp_5

```
typedef ex(*) GiNaC::eval_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.77 evalf_funcp_5

```
typedef ex(*) GiNaC::evalf_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.78 conjugate_funcp_5

```
typedef ex(*) GiNaC::conjugate_funcp_5(const ex &, const ex &, const ex &, const ex &, const
ex &)
```

7.1.1.79 real_part_funcp_5

```
typedef ex(*) GiNaC::real_part_funcp_5(const ex &, const ex &, const ex &, const ex &, const
ex &)
```

7.1.1.80 imag_part_funcp_5

```
typedef ex(*) GiNaC::imag_part_funcp_5(const ex &, const ex &, const ex &, const ex &, const
ex &)
```

7.1.1.81 expand_funcp_5

```
typedef ex(*) GiNaC::expand_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex
&, unsigned)
```

7.1.1.82 derivative_funcp_5

```
typedef ex(*) GiNaC::derivative_funcp_5(const ex &, const ex &, const ex &, const ex &, const
ex &, unsigned)
```

7.1.1.83 expl_derivative_funcp_5

```
typedef ex(*) GiNaC::expl_derivative_funcp_5(const ex &, const ex &, const ex &, const ex &,
const ex &, const symbol &)
```

7.1.1.84 power_funcp_5

```
typedef ex(*) GiNaC::power_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &)
```

7.1.1.85 series_funcp_5

```
typedef ex(*) GiNaC::series_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex
&, const relational &, int, unsigned)
```

7.1.1.86 print_funcp_5

```
typedef void(*) GiNaC::print_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex
&, const print_context &)
```

7.1.1.87 info_funcp_5

```
typedef bool(*) GiNaC::info_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex
&, unsigned)
```

7.1.1.88 eval_funcp_6

```
typedef ex(*) GiNaC::eval_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &)
```

7.1.1.89 evalf_funcp_6

```
typedef ex(*) GiNaC::evalf_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &)
```

7.1.1.90 conjugate_funcp_6

```
typedef ex(*) GiNaC::conjugate_funcp_6(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &)
```

7.1.1.91 real_part_funcp_6

```
typedef ex(*) GiNaC::real_part_funcp_6(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &)
```

7.1.1.92 imag_part_funcp_6

```
typedef ex(*) GiNaC::imag_part_funcp_6(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &)
```

7.1.1.93 expand_funcp_6

```
typedef ex(*) GiNaC::expand_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, unsigned)
```

7.1.1.94 derivative_funcp_6

```
typedef ex(*) GiNaC::derivative_funcp_6(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, unsigned)
```

7.1.1.95 expl_derivative_funcp_6

```
typedef ex(*) GiNaC::expl_derivative_funcp_6(const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const symbol &)
```

7.1.1.96 power_funcp_6

```
typedef ex(*) GiNaC::power_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &)
```

7.1.1.97 series_funcp_6

```
typedef ex(*) GiNaC::series_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const relational &, int, unsigned)
```

7.1.1.98 print_funcp_6

```
typedef void(*) GiNaC::print_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const print_context &)
```

7.1.1.99 info_funcp_6

```
typedef bool(*) GiNaC::info_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, unsigned)
```

7.1.1.100 eval_funcp_7

```
typedef ex(*) GiNaC::eval_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &)
```

7.1.1.101 evalf_funcp_7

```
typedef ex(*) GiNaC::evalf_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &)
```

7.1.1.102 conjugate_funcp_7

```
typedef ex(*) GiNaC::conjugate_funcp_7(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &)
```

7.1.1.103 real_part_funcp_7

```
typedef ex(*) GiNaC::real_part_funcp_7(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &)
```

7.1.1.104 imag_part_funcp_7

```
typedef ex(*) GiNaC::imag_part_funcp_7(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &)
```

7.1.1.105 expand_funcp_7

```
typedef ex(*) GiNaC::expand_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, unsigned)
```

7.1.1.106 derivative_funcp_7

```
typedef ex(*) GiNaC::derivative_funcp_7(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, unsigned)
```

7.1.1.107 expl_derivative_funcp_7

```
typedef ex(*) GiNaC::expl_derivative_funcp_7(const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const symbol &)
```

7.1.1.108 power_funcp_7

```
typedef ex(*) GiNaC::power_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &)
```

7.1.1.109 series_funcp_7

```
typedef ex(*) GiNaC::series_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const relational &, int, unsigned)
```

7.1.1.110 print_funcp_7

```
typedef void(*) GiNaC::print_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const print_context &)
```

7.1.1.111 info_funcp_7

```
typedef bool(*) GiNaC::info_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, unsigned)
```

7.1.1.112 eval_funcp_8

```
typedef ex(*) GiNaC::eval_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &)
```

7.1.1.113 evalf_funcp_8

```
typedef ex(*) GiNaC::evalf_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &)
```

7.1.1.114 conjugate_funcp_8

```
typedef ex(*) GiNaC::conjugate_funcp_8(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &)
```

7.1.1.115 real_part_funcp_8

```
typedef ex(*) GiNaC::real_part_funcp_8(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &)
```

7.1.1.116 imag_part_funcp_8

```
typedef ex(*) GiNaC::imag_part_funcp_8(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &)
```

7.1.1.117 expand_funcp_8

```
typedef ex(*) GiNaC::expand_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.118 derivative_funcp_8

```
typedef ex(*) GiNaC::derivative_funcp_8(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.119 expl_derivative_funcp_8

```
typedef ex(*) GiNaC::expl_derivative_funcp_8(const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const symbol &)
```

7.1.1.120 power_funcp_8

```
typedef ex(*) GiNaC::power_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &)
```

7.1.1.121 series_funcp_8

```
typedef ex(*) GiNaC::series_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const relational &, int, unsigned)
```

7.1.1.122 print_funcp_8

```
typedef void(*) GiNaC::print_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const print_context &)
```

7.1.1.123 info_funcp_8

```
typedef bool(*) GiNaC::info_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.124 eval_funcp_9

```
typedef ex(*) GiNaC::eval_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &)
```

7.1.1.125 evalf_funcp_9

```
typedef ex(*) GiNaC::evalf_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &)
```

7.1.1.126 conjugate_funcp_9

```
typedef ex(*) GiNaC::conjugate_funcp_9(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.127 real_part_funcp_9

```
typedef ex(*) GiNaC::real_part_funcp_9(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.128 imag_part_funcp_9

```
typedef ex(*) GiNaC::imag_part_funcp_9(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.129 expand_funcp_9

```
typedef ex(*) GiNaC::expand_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.130 derivative_funcp_9

```
typedef ex(*) GiNaC::derivative_funcp_9(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.131 expl_derivative_funcp_9

```
typedef ex(*) GiNaC::expl_derivative_funcp_9(const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
```

7.1.1.132 power_funcp_9

```
typedef ex(*) GiNaC::power_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.133 series_funcp_9

```
typedef ex(*) GiNaC::series_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
```

7.1.1.134 print_funcp_9

```
typedef void(*) GiNaC::print_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const print_context &)
```

7.1.1.135 info_funcp_9

```
typedef bool(*) GiNaC::info_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.136 eval_funcp_10

```
typedef ex(*) GiNaC::eval_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.137 evalf_funcp_10

```
typedef ex(*) GiNaC::evalf_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.138 conjugate_funcp_10

```
typedef ex(*) GiNaC::conjugate_funcp_10(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.139 real_part_funcp_10

```
typedef ex(*) GiNaC::real_part_funcp_10(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.140 imag_part_funcp_10

```
typedef ex(*) GiNaC::imag_part_funcp_10(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.141 expand_funcp_10

```
typedef ex(*) GiNaC::expand_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.142 derivative_funcp_10

```
typedef ex(*) GiNaC::derivative_funcp_10(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.143 expl_derivative_funcp_10

```
typedef ex(*) GiNaC::expl_derivative_funcp_10(const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
```

7.1.1.144 power_funcp_10

```
typedef ex(*) GiNaC::power_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.145 series_funcp_10

```
typedef ex(*) GiNaC::series_funcp_10(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int,
unsigned)
```

7.1.1.146 print_funcp_10

```
typedef void(*) GiNaC::print_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
```

7.1.1.147 info_funcp_10

```
typedef bool(*) GiNaC::info_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.148 eval_funcp_11

```
typedef ex(*) GiNaC::eval_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.149 evalf_funcp_11

```
typedef ex(*) GiNaC::evalf_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.150 conjugate_funcp_11

```
typedef ex(*) GiNaC::conjugate_funcp_11(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.151 real_part_funcp_11

```
typedef ex(*) GiNaC::real_part_funcp_11(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.152 imag_part_funcp_11

```
typedef ex(*) GiNaC::imag_part_funcp_11(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.153 expand_funcp_11

```
typedef ex(*) GiNaC::expand_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.154 derivative_funcp_11

```
typedef ex(*) GiNaC::derivative_funcp_11(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.155 expl_derivative_funcp_11

```
typedef ex(*) GiNaC::expl_derivative_funcp_11(const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
symbol &)
```

7.1.1.156 power_funcp_11

```
typedef ex(*) GiNaC::power_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.157 series_funcp_11

```
typedef ex(*) GiNaC::series_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &,
int, unsigned)
```

7.1.1.158 print_funcp_11

```
typedef void(*) GiNaC::print_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context
&)
```

7.1.1.159 info_funcp_11

```
typedef bool(*) GiNaC::info_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
```

7.1.1.160 eval_funcp_12

```
typedef ex(*) GiNaC::eval_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.161 evalf_funcp_12

```
typedef ex(*) GiNaC::evalf_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.162 conjugate_funcp_12

```
typedef ex(*) GiNaC::conjugate_funcp_12(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.163 real_part_funcp_12

```
typedef ex(*) GiNaC::real_part_funcp_12(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.164 imag_part_funcp_12

```
typedef ex(*) GiNaC::imag_part_funcp_12(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
```

7.1.1.165 expand_funcp_12

```
typedef ex(*) GiNaC::expand_funcp_12(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
unsigned)
```

7.1.1.166 derivative_funcp_12

```
typedef ex(*) GiNaC::derivative_funcp_12(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
unsigned)
```

7.1.1.167 expl_derivative_funcp_12

```
typedef ex(*) GiNaC::expl_derivative_funcp_12(const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex
&, const symbol &)
```

7.1.1.168 power_funcp_12

```
typedef ex(*) GiNaC::power_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &)
```

7.1.1.169 series_funcp_12

```
typedef ex(*) GiNaC::series_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
relational &, int, unsigned)
```

7.1.1.170 print_funcp_12

```
typedef void(*) GiNaC::print_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
print_context &)
```

7.1.1.171 info_funcp_12

```
typedef bool(*) GiNaC::info_funcp_12(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
unsigned)
```

7.1.1.172 eval_funcp_13

```
typedef ex(*) GiNaC::eval_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex
&)
```

7.1.1.173 evalf_funcp_13

```
typedef ex(*) GiNaC::evalf_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &)
```

7.1.1.174 conjugate_funcp_13

```
typedef ex(*) GiNaC::conjugate_funcp_13(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &)
```

7.1.1.175 real_part_funcp_13

```
typedef ex(*) GiNaC::real_part_funcp_13(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &)
```

7.1.1.176 imag_part_funcp_13

```
typedef ex(*) GiNaC::imag_part_funcp_13(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &)
```

7.1.1.177 expand_funcp_13

```
typedef ex(*) GiNaC::expand_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, unsigned)
```

7.1.1.178 derivative_funcp_13

```
typedef ex(*) GiNaC::derivative_funcp_13(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, unsigned)
```

7.1.1.179 expl_derivative_funcp_13

```
typedef ex(*) GiNaC::expl_derivative_funcp_13(const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const symbol &)
```

7.1.1.180 power_funcp_13

```
typedef ex(*) GiNaC::power_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &)
```

7.1.1.181 series_funcp_13

```
typedef ex(*) GiNaC::series_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, const relational &, int, unsigned)
```

7.1.1.182 print_funcp_13

```
typedef void(*) GiNaC::print_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, const print_context &)
```

7.1.1.183 info_funcp_13

```
typedef bool(*) GiNaC::info_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, unsigned)
```

7.1.1.184 eval_funcp_14

```
typedef ex(*) GiNaC::eval_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &)
```

7.1.1.185 evalf_funcp_14

```
typedef ex(*) GiNaC::evalf_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &)
```

7.1.1.186 conjugate_funcp_14

```
typedef ex(*) GiNaC::conjugate_funcp_14(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &)
```

7.1.1.187 real_part_funcp_14

```
typedef ex(*) GiNaC::real_part_funcp_14(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &)
```

7.1.1.188 imag_part_funcp_14

```
typedef ex(*) GiNaC::imag_part_funcp_14(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &)
```

7.1.1.189 expand_funcp_14

```
typedef ex(*) GiNaC::expand_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, unsigned)
```

7.1.1.190 derivative_funcp_14

```
typedef ex(*) GiNaC::derivative_funcp_14(const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, unsigned)
```

7.1.1.191 expl_derivative_funcp_14

```
typedef ex(*) GiNaC::expl_derivative_funcp_14(const ex &, const ex &, const ex &, const ex &,
const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const symbol &)
```

7.1.1.192 power_funcp_14

```
typedef ex(*) GiNaC::power_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const ex &)
```

7.1.1.193 series_funcp_14

```
typedef ex(*) GiNaC::series_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const relational &, int, unsigned)
```

7.1.1.194 print_funcp_14

```
typedef void(*) GiNaC::print_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, const print_context &)
```

7.1.1.195 info_funcp_14

```
typedef bool(*) GiNaC::info_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex
&, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const
ex &, const ex &, unsigned)
```

7.1.1.196 eval_funcp_exvector

```
typedef ex(*) GiNaC::eval_funcp_exvector(const exvector &)
```

7.1.1.197 evalf_funcp_exvector

```
typedef ex(*) GiNaC::evalf_funcp_exvector(const exvector &)
```

7.1.1.198 conjugate_funcp_exvector

```
typedef ex(*) GiNaC::conjugate_funcp_exvector(const exvector &)
```

7.1.1.199 real_part_funcp_exvector

```
typedef ex(*) GiNaC::real_part_funcp_exvector(const exvector &)
```

7.1.1.200 imag_part_funcp_exvector

```
typedef ex(*) GiNaC::imag_part_funcp_exvector(const exvector &)
```

7.1.1.201 expand_funcp_exvector

```
typedef ex(*) GiNaC::expand_funcp_exvector(const exvector &, unsigned)
```

7.1.1.202 derivative_funcp_exvector

```
typedef ex(*) GiNaC::derivative_funcp_exvector(const exvector &, unsigned)
```

7.1.1.203 expl_derivative_funcp_exvector

```
typedef ex(*) GiNaC::expl_derivative_funcp_exvector(const exvector &, const symbol &)
```

7.1.1.204 power_funcp_exvector

```
typedef ex(*) GiNaC::power_funcp_exvector(const exvector &, const ex &)
```

7.1.1.205 series_funcp_exvector

```
typedef ex(*) GiNaC::series_funcp_exvector(const exvector &, const relational &, int, unsigned)
```

7.1.1.206 print_funcp_exvector

```
typedef void(*) GiNaC::print_funcp_exvector(const exvector &, const print_context &)
```

7.1.1.207 info_funcp_exvector

```
typedef bool(*) GiNaC::info_funcp_exvector(const exvector &, unsigned)
```

7.1.1.208 exhashmap

```
template<typename T, class Hash = std::hash<ex>, class KeyEqual = std::equal_to<ex>, class
Allocator = std::allocator<std::pair<const ex, T>>>
using GiNaC::exhashmap = std::unordered_map<ex, T, Hash, KeyEqual, Allocator>
```

7.1.1.209 spmap

```
typedef std::map<spmapkey, ex> GiNaC::spmap
```

7.1.1.210 lookup_map

```
typedef map<error_and_integral, ex, error_and_integral_is_less> GiNaC::lookup_map
```

7.1.1.211 lst

```
typedef container< std::list > GiNaC::lst
```

7.1.1.212 uintvector

```
typedef std::vector<std::size_t> GiNaC::uintvector
```

7.1.1.213 unsignedvector

```
typedef std::vector<unsigned> GiNaC::unsignedvector
```

7.1.1.214 exvectorvector

```
typedef std::vector<exvector> GiNaC::exvectorvector
```

7.1.1.215 sym_desc_vec

```
typedef std::vector<sym_desc> GiNaC::sym_desc_vec
```

7.1.1.216 digits_changed_callback

```
typedef void(*) GiNaC::digits_changed_callback(long)
```

Function pointer to implement callbacks in the case 'Digits' gets changed.

Main purpose of such callbacks is to adjust look-up tables of certain functions to the new precision. Parameter contains the signed difference between new Digits and old Digits.

7.1.1.217 print_context_class_info

```
typedef class_info<print_context_options> GiNaC::print_context_class_info
```

7.1.1.218 registered_class_info

```
typedef class_info<registered_class_options> GiNaC::registered_class_info
```

7.1.2 Enumeration Type Documentation

7.1.2.1 anonymous enum

```
anonymous enum
```

Enumerator

callback_registered	

7.1.3 Function Documentation

7.1.3.1 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [1/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            add ,
            expairseq ,
            print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_csrc > &::do_print_csrc.  print_func< print_tree > &::do_↩
print_tree.  print_func< print_python_repr > &::do_print_python_repr )
```

References GiNaC::add::do_print(), GiNaC::add::do_print_csrc(), GiNaC::add::do_print_latex(), GiNaC::add::do_print_python_repr(), GiNaC::basic::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

Referenced by GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), print_func< print_dflt >(), print_func< print_dflt >(), and print_func< print_dflt >().

7.1.3.2 GINAC_BIND_UNARCHIVER() [1/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            add )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.3 GINAC_DECLARE_UNARCHIVER() [1/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            add )
```

References GINAC_DECLARE_UNARCHIVER().

Referenced by GINAC_DECLARE_UNARCHIVER(),

GINAC_DECLARE_UNARCHIVER(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.4 write_unsigned()

```
void GiNaC::write_unsigned (
             std::ostream & os,
             unsigned val)  [static]
```

Write unsigned integer quantity to stream.

References write_unsigned().

Referenced by GiNaC::archive::operator<<, GiNaC::archive_node::operator<<, and write_unsigned().

7.1.3.5 read_unsigned()

```
unsigned GiNaC::read_unsigned (
             std::istream & is)  [static]
```

Read unsigned integer quantity from stream.

References read_unsigned().

Referenced by GiNaC::archive::operator>>, GiNaC::archive_node::operator>>, and read_unsigned().

7.1.3.6 operator<<() [1/16]

```
std::ostream & GiNaC::operator<< (
             std::ostream & os,
             const archive_node & n)
```

Write archive_node to binary data stream.

Referenced by operator<<(), operator<<(), operator<<(), operator<<(), and operator<<().

7.1.3.7 operator<<() [2/16]

```
std::ostream & GiNaC::operator<< (
             std::ostream & os,
             const archive & ar)
```

Write archive to binary data stream.

7.1.3.8 operator>>() [1/3]

```
std::istream & GiNaC::operator>> (
            std::istream & is,
            archive_node & n)
```

Read archive_node from binary data stream.

Referenced by operator>>().

7.1.3.9 operator>>() [2/3]

```
std::istream & GiNaC::operator>> (
            std::istream & is,
            archive & ar)
```

Read archive from binary data stream.

7.1.3.10 find_factory_fcn()

```
synthesize_func GiNaC::find_factory_fcn (
            const std::string & name)  [static]
```

References GiNaC::unarchive_table_t::find(), and find_factory_fcn().

Referenced by find_factory_fcn(), and GiNaC::archive_node::unarchive().

7.1.3.11 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [2/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            basic ,
            void ,
            print_func< print_context > &::do_print.  print_func< print_tree > &::do_print↩
_tree.  print_func< print_python_repr > &::do_print_python_repr )
```

basic copy constructor: implicitly assumes that the other class is of the exact same type (as it's used by duplicate()), so it can copy the tinfo_key and the hash value.

References GiNaC::basic::do_print(), GiNaC::basic::do_print_python_repr(), GiNaC::basic::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.12 is_a() [1/3]

```
template<class T>
bool GiNaC::is_a (
            const basic & obj)  [inline]
```

Check if obj is a T, including base classes.

References is_a().

Referenced by abs_power(), GiNaC::matrix::add_indexed(), atan2_eval(), atan_series(), atanh_series(), base_and_index(), beta_series(), canonicalize_clifford(), GiNaC::clifford::clifford(), clifford_max_label(), clifford_moebius_map(), clifford_moebius_map(), clifford_prime(), clifford_star_bar(), clifford_to_lst(), clifford_unit(), GiNaC::basic::collect(), collect_symbols(), color_d(), color_f(), color_T(), color_trace(), GiNaC::expairseq::construct_from_2_ex(), GiNaC::cliffordunit::contract_with(), GiNaC::diracgamma::contract_with(), GiNaC::matrix::contract_with(), GiNaC::spinmetric::contract_with(), GiNaC::su3d::contract_with(), GiNaC::su3f::contract_with(), GiNaC::su3t::contract_with(), GiNaC::tensdelta::contract_with(), GiNaC::tensepsilon::contract_with(), GiNaC::tensmetric::contract_with(), convert_H_to_Li(), csgn_power(), delta_tensor(), GiNaC::power::derivative(), dirac_gamma(), dirac_trace(), GiNaC::add::do_print_csrc(), GiNaC::mul::do_print_csrc(), GiNaC::numeric::do_print_csrc(), GiNaC::power::do_print_csrc(), epsilon_tensor(), epsilon_tensor(), GiNaC::function::eval(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::matrix::eval_indexed(), GiNaC::minkmetric::eval_indexed(), GiNaC::spinmetric::eval_indexed(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::tensdelta::eval_indexed(), GiNaC::tensepsilon::eval_indexed(), GiNaC::tensmetric::eval_indexed(), GiNaC::integral::eval_integ(), GiNaC::clifford::eval_ncmul(), GiNaC::color::eval_ncmul(), GiNaC::Gt::evalf(), GiNaC::add::evalm(), GiNaC::mul::evalm(), GiNaC::ncmul::evalm(), GiNaC::power::evalm(), GiNaC::expairseq::expairseq(), GiNaC::expairseq::expairseq(), GiNaC::integral::expand(), GiNaC::mul::expand(), GiNaC::power::expand(), expand_dummy_sum(), find_common_factor(), fsolve(), G2_eval(), G2_evalf(), G3_eval(), G3_evalf(), gcd(), gcd_pf_pow(), get_all_dummy_indices(), get_all_dummy_indices_safely(), get_clifford_comp(), get_dim_uint(), get_first_symbol(), GiNaC::clifford::get_metric(), GINAC_BIND_UNARCHIVER(), GiNaC::Gt::Gt(), GiNaC::Gt::Gt(), H_deriv(), H_eval(), H_evalf(), H_print_latex(), GiNaC::make_flat_inserter::handle_factor(), GiNaC::mul::has(), GiNaC::power::has(), hasindex(), haswild(), GiNaC::symbolset::insert_symbols(), GiNaC::integral::integral(), is_a(), is_a(), is_dirac_slash(), is_dummy_pair(), GiNaC::idx::is_dummy_pair_same_type(), GiNaC::container< C >::is_equal_same_type(), GiNaC::fderivative::is_equal_same_type(), GiNaC::function::is_equal_same_type(), GiNaC::structure< T, ComparisonPolicy >::is_equal_same_type(), GiNaC::symbol::is_equal_same_type(), iterated_integral_evalf_impl(), lcmcoeff(), Li_deriv(), Li_eval(), Li_evalf(), Li_print_latex(), Li_series(), log_series(), lorentz_eps(), lorentz_g(), lst_to_clifford(), lst_to_matrix(), GiNaC::expairseq::make_flat(), GiNaC::expairseq::make_flat(), GiNaC::expairseq::map(), GiNaC::clifford::match_same_type(), GiNaC::color::match_same_type(), GiNaC::fderivative::match_same_type(), GiNaC::function::match_same_type(), GiNaC::idx::match_same_type(), GiNaC::spinidx::match_same_type(), GiNaC::varidx::match_same_type(), metric_tensor(), minimal_dim(), multiply_lcm(), GiNaC::basic::normal(), GiNaC::ex_base_is_less::operator()(), GiNaC::is_not_a_clifford::operator()(), GiNaC::spmapkey::operator<(), GiNaC::spmapkey::operator==(), Order_series(), GiNaC::function::print(), GiNaC::idx::print_index(), print_real_number(), GiNaC::indexed::printindices(), GiNaC::pseries::pseries(), GiNaC::pseries::pseries(), GiNaC::symbol::read_archive(), remove_dirac_ONE(), rename_dummy_indices_uniquely(), GiNaC::tensor::replace_contr_index(), replace_with_symbol(), GiNaC::indexed::reposition_dummy_indices, GiNaC::indexed::return_type(), S_evalf(), GiNaC::clifford::same_metric(), GiNaC::matrix::scalar_mul_indexed(), GiNaC::mul::series(), GiNaC::power::series(), GiNaC::pseries::series(), GiNaC::symbol::series(), GiNaC::Gt::setArgs(), GiNaC::indexed::simplify_indexed, GiNaC::indexed::simplify_indexed_product, spinor_metric(), GiNaC::spmapkey::spmapkey(), sqrfree(), GiNaC::clifford::subs(), GiNaC::container< C >::subs(), GiNaC::idx::subs(), subsvalue(), tan_series(), tanh_series(), GiNaC::Gt::to_integer(), GiNaC::Gt::to_numeric(), GiNaC::indexed::validate(), zeta1_print_latex(), and zeta2_print_latex().

7.1.3.13 is_exactly_a() [1/2]

```
template<class T>
bool GiNaC::is_exactly_a (
            const basic & obj)  [inline]
```

Check if obj is a T, not including base classes.

References is_exactly_a().

Referenced by abs_eval(), abs_evalf(), abs_expand(), acos_conjugate(), acos_evalf(), acosh_conjugate(), acosh_evalf(), adaptivesimpson(), GiNaC::symmetry::add(), GiNaC::mul::algebraic_subs_mul(), GiNaC::ncmul::append_factors(), asin_conjugate(), asin_evalf(), asinh_conjugate(), asinh_evalf(), atan2_evalf(), atan_conjugate(), atan_evalf(), atanh_conjugate(), atanh_evalf(), beta_evalf(), binomial_eval(), GiNaC::mul::can_be_further_expanded(), GiNaC::mul::can_make_flat(), GiNaC::symmetry::canonicalize, canonicalize_clifford(), GiNaC::Gt_detail::TransformExpressionWithCa GiNaC::power::coeff(), GiNaC::pseries::coeff(), collect_common_factors(), collect_symbols(), color_trace(), GiNaC::add::combine_ex_with_coeff_to_pair(), GiNaC::expairseq::combine_ex_with_coeff_to_pair(), GiNaC::mul::combine_ex_with_ GiNaC::expairseq::combine_overall_coeff(), GiNaC::expairseq::combine_overall_coeff(), GiNaC::mul::combine_overall_coeff(), GiNaC::mul::combine_overall_coeff(), GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::expairseq::combine_pair_with_coeff_ GiNaC::mul::combine_pair_with_coeff_to_pair(), conjugate_evalf(), GiNaC::expairseq::construct_from_2_ex(), GiNaC::expairseq::construct_from_expairseq_ex(), GiNaC::su3d::contract_with(), GiNaC::su3f::contract_with(), GiNaC::su3t::contract_with(), GiNaC::tensepsilon::contract_with(), cos_eval(), cos_evalf(), cosh_eval(), cosh_evalf(), GiNaC::ncmul::count_factors(), csgn_eval(), csgn_evalf(), decomp_rational(), GiNaC::power::degree(), dirac_trace(), divide(), divide_in_z(), GiNaC::mul::do_print_latex(), GiNaC::power::do_print_latex(), GiNaC::add::eval(), GiNaC::indexed::eval(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::integral::evalf(), GiNaC::power::evalf(), GiNaC::power::evalm(), exp_evalf(), exp_expand(), GiNaC::expair::expair(), GiNaC::expairseq::expair_needs_further_processing(), GiNaC::mul::expair_needs_further_processing(), GiNaC::indexed::expand(), GiNaC::mul::expand(), GiNaC::ncmul::expand(), GiNaC::power::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), GiNaC::power::expand_mul(), factorial_eval(), factorial_print_dflt_latex(), find_common_factor(), find_variant_indices(), frac_cancel(), gcd(), gcd_pf_mul(), gcd_pf_pow(), get_first_symbol(), GiNaC::numeric::has(), heur_gcd_z(), idx_symmetrization(), imag_part_evalf(), GiNaC::Gt_detail::TransformExpressionWithCache< Type >::impl(), GiNaC::add::integer_content(), GiNaC::mul::integer_content(), GiNaC::expairseq::is_canonical(), GiNaC::expair::is_canonical_numeric(), GiNaC::idx::is_dim_numeri GiNaC::idx::is_dim_symbolic(), GiNaC::idx::is_dummy_pair_same_type(), GiNaC::constant::is_equal_same_type(), GiNaC::numeric::is_equal_same_type(), is_exactly_a(), GiNaC::idx::is_numeric(), GiNaC::idx::is_symbolic(), is_the_function(), lcm(), lcmcoeff(), GiNaC::power::ldegree(), lgamma_conjugate(), lgamma_evalf(), Li2_conjugate(), Li2_evalf(), log_conjugate(), log_evalf(), log_expand(), lsolve(), GiNaC::expairseq::make_flat(), GiNaC::basic::match(), GiNaC::expairseq::match(), GiNaC::matrix::match_same_type(), GiNaC::relational::match_same_type(), GiNaC::add::max_coefficien GiNaC::mul::max_coefficient(), minimal_dim(), multiply_lcm(), number_of_type(), GiNaC::relational::operator safe_bool(), GiNaC::sy_is_less::operator()(), GiNaC::sy_swap::operator()(), GiNaC::basic::operator[ ](), GiNaC::basic::operator[ ](), Order_eval(), Order_power(), GiNaC::matrix::pivot(), GiNaC::matrix::pow(), prem(), GiNaC::idx::print_index(), product_to_exvector(), psi1_evalf(), psi2_evalf(), quo(), real_part_evalf(), rem(), rename_dummy_indices(), rename_dummy_indices_uniquely(), GiNaC::indexed::reposition_dummy_indices, GiNaC::add::series(), GiNaC::power::series(), GiNaC::indexed::simplify_indexed, GiNaC::indexed::simplify_indexed_product, sin_eval(), sin_evalf(), sinh_eval(), sinh_evalf(), GiNaC::add::smod(), GiNaC::mul::smod(), GiNaC::add::split_ex_to_pair(), GiNaC::mul::split_ex_to_pair(), sprem(), sqrfree(), sr_gcd(), step_eval(), step_evalf(), GiNaC::expairseq::subs(), GiNaC::expairseq::subschildren(), GiNaC::symminfo::symminfo(), tan_eval(), tan_evalf(), tanh_eval(), tanh_evalf(), tgamma_evalf(), GiNaC::power::to_polynomial(), tryfactsubs(), GiNaC::indexed::validate(), zeta1_deriv(), zeta1_eval(), zeta1_evalf(), zeta2_deriv(), zeta2_eval(), and zeta2_evalf().

7.1.3.14 dynallocate() [1/2]

```
template<class B, typename... Args>
B & GiNaC::dynallocate (
            Args &&... args) [inline]
```

Constructs a new (class basic or derived) B object on the heap.

This function picks the object's ctor based on the given argument types.

This helps the constructor of ex from basic (or a derived class B) because then the constructor doesn't have to duplicate the object onto the heap. See ex::construct_from_basic(const basic &) for more information.

References dynallocate(), and GiNaC::status_flags::dynallocated.

Referenced by abs_expand(), GiNaC::numeric::add_dyn(), antisymmetric2(), antisymmetric3(), antisymmetric4(), base_and_index(), clifford_moebius_map(), clifford_star_bar(), clifford_unit(), GiNaC::add::coeff(), GiNaC::mul::coeff(), GiNaC::ncmul::coeff(), GiNaC::basic::collect(), color_d(), color_f(), color_ONE(), color_T(), color_trace(), GiNaC::add::combine_ex_with_coeff_to_pair(), GiNaC::diracgammaL::conjugate(), GiNaC::diracgammaR::conjugate(), GiNaC::integral::conjugate(), GiNaC::ncmul::conjugate(), GiNaC::power::conjugate(), GiNaC::pseries::conjugate(), GiNaC::ex::construct_from_double(), GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), GiNaC::ex::construct_from_ulonglong(), delta_tensor(), GiNaC::add::derivative(), GiNaC::mul::derivative(), GiNaC::ncmul::derivative(), GiNaC::power::derivative(), diag_matrix(), diag_matrix(), dirac_gamma(), dirac_gamma5(), dirac_gammaL(), dirac_gammaR(), dirac_ONE(), dirac_slash(), GiNaC::numeric::div_dyn(), divide(), divide_in_z(), dynallocate(), dynallocate(), epsilon_tensor(), epsilon_tensor(), GiNaC::add::eval(), GiNaC::expairseq::eval(), GiNaC::mul::eval(), GiNaC::ncmul::eval(), GiNaC::power::eval(), GiNaC::pseries::eval_integ(), GiNaC::clifford::eval_ncmul(), GiNaC::Gt::evalf(), GiNaC::integral::evalf(), GiNaC::mul::evalf(), GiNaC::power::evalf(), GiNaC::pseries::evalf(), GiNaC::add::evalm(), GiNaC::mul::evalm(), GiNaC::ncmul::evalm(), GiNaC::power::evalm(), GiNaC::pseries::evalm(), exadd(), exminus(), exmul(), exp_expand(), GiNaC::add::expand(), GiNaC::integral::expand(), GiNaC::mul::expand(), GiNaC::ncmul::expand(), GiNaC::power::expand(), GiNaC::pseries::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), GiNaC::power::expand_mul(), find_common_factor(), frac_cancel(), gcd_pf_mul(), get_clifford_comp(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GiNaC::ncmul::hold_ncmul, GiNaC::add::imag_part(), GiNaC::pseries::imag_part(), index0(), index1(), index2(), index3(), interpolate(), GiNaC::library_init::library_init(), log_series(), lorentz_eps(), lorentz_g(), lst_to_matrix(), GiNaC::power::map(), GiNaC::relational::map(), metric_tensor(), GiNaC::numeric::mul_dyn(), GiNaC::pseries::mul_series(), multiply_lcm(), GiNaC::basic::normal(), GiNaC::mul::normal(), GiNaC::numeric::normal(), GiNaC::power::normal(), GiNaC::pseries::normal(), GiNaC::symbol::normal(), not_symmetric(), pow(), pow(), GiNaC::function::power(), GiNaC::pseries::power_const(), GiNaC::numeric::power_dyn(), quo(), GiNaC::add::real_part(), GiNaC::pseries::real_part(), GiNaC::add::recombine_pair_to_ex(), GiNaC::mul::recombine_pair_to_ex(), reduced_matrix(), GiNaC::ncmul::reeval_ncmul, rem(), rename_dummy_indices_uniquely(), replace_with_symbol(), replace_with_symbol(), GiNaC::integral::series(), GiNaC::mul::series(), GiNaC::indexed::simplify_indexed, GiNaC::add::smod(), GiNaC::mul::smod(), spinor_metric(), GiNaC::add::split_ex_to_pair(), GiNaC::numeric::sub_dyn(), sub_matrix(), GiNaC::power::subs(), GiNaC::pseries::subs(), symbolic_matrix(), symm(), symmetric2(), symmetric3(), symmetric4(), GiNaC::ncmul::thiscontainer(), GiNaC::ncmul::thiscontainer(), GiNaC::add::thisexpairseq(), GiNaC::add::thisexpairseq(), GiNaC::mul::thisexpairseq(), GiNaC::mul::thisexpairseq(), and unit_matrix().

7.1.3.15 dynallocate() [2/2]

```
template<class B>
B & GiNaC::dynallocate (
             std::initializer_list< ex > il)  [inline]
```

Constructs a new (class basic or derived) B object on the heap.

This function is needed for GiNaC classes which have public ctors from initializer lists of expressions (which are not a type and not captured by the variadic template version).

References dynallocate(), and GiNaC::status_flags::dynallocated.

7.1.3.16 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [3/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             clifford ,
             indexed ,
             print_func< print_dflt > &::do_print_dflt.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_tree > &::do_print_tree )
```

References GiNaC::clifford::do_print_dflt(), GiNaC::clifford::do_print_latex(), GiNaC::clifford::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.17 print_func< print_dflt >() [1/3]

```
GiNaC::print_func< print_dflt > (
            &diracone::do_print ) &
```

References GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.18 GINAC_BIND_UNARCHIVER() [2/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            clifford )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.19 GINAC_BIND_UNARCHIVER() [3/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            cliffordunit )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.20 GINAC_BIND_UNARCHIVER() [4/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            diracone )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.21 GINAC_BIND_UNARCHIVER() [5/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            diracgamma )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.22 GINAC_BIND_UNARCHIVER() [6/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            diracgamma5 )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.23 GINAC_BIND_UNARCHIVER() [7/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            diracgammaL )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.24 GINAC_BIND_UNARCHIVER() [8/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            diracgammaR )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.25 is_dirac_slash()

```
bool GiNaC::is_dirac_slash (
            const ex & seq0)  [static]
```

References is_a(), and is_dirac_slash().

Referenced by GiNaC::clifford::do_print_dflt(), GiNaC::clifford::do_print_latex(), and is_dirac_slash().

7.1.3.26 base_and_index()

```
void GiNaC::base_and_index (
            const ex & c,
            ex & b,
            ex & i)  [static]
```

This function decomposes gamma~mu -> (1, mu) and a\ -> (a.ix, ix).

References _ex0, _ex1, base_and_index(), dynallocate(), ex_to(), GINAC_ASSERT, GiNaC::indexed::indexed(), is_a(), and GiNaC::varidx::toggle_variance().

Referenced by base_and_index(), canonicalize_clifford(), GiNaC::diracgamma::contract_with(), and dirac_trace().

7.1.3.27 dirac_ONE()

```
ex GiNaC::dirac_ONE (
            unsigned char rl = 0)
```

Create a Clifford unity object.

Parameters

rl	Representation label

Returns

newly constructed object

References GiNaC::clifford::clifford(), dirac_ONE(), and dynallocate().

Referenced by canonicalize_clifford(), clifford_to_lst(), GiNaC::add::coeff(), GiNaC::cliffordunit::contract_with(), GiNaC::diracgamma::contract_with(), dirac_ONE(), dirac_trace(), GiNaC::clifford::eval_ncmul(), and lst_to_clifford().

7.1.3.28 get_dim_uint()

```
unsigned GiNaC::get_dim_uint (
            const ex & e)  [static]
```

References ex_to(), get_dim_uint(), GiNaC::ex::info(), is_a(), and GiNaC::info_flags::posint.

Referenced by clifford_unit(), get_dim_uint(), and lst_to_clifford().

7.1.3.29 clifford_unit()

```
ex GiNaC::clifford_unit (
            const ex & mu,
            const ex & metr,
            unsigned char rl = 0)
```

Create a Clifford unit object.

Parameters

mu	Index (must be of class varidx or a derived class)
metr	Metric (should be indexed, tensmetric or a derived class, or a matrix)
rl	Representation label

Returns

newly constructed Clifford unit object

References GiNaC::clifford::clifford(), clifford_unit(), GiNaC::matrix::cols(), dynallocate(), ex_to(), get_dim_uint(), GiNaC::ex::get_free_indices(), GiNaC::indexed::indexed(), is_a(), not_symmetric(), GiNaC::matrix::rows(), and symmetric2().

Referenced by clifford_moebius_map(), clifford_unit(), and lst_to_clifford().

7.1.3.30 dirac_gamma()

```
ex GiNaC::dirac_gamma (
            const ex & mu,
            unsigned char rl = 0)
```

Create a Dirac gamma object.

Parameters

mu	Index (must be of class varidx or a derived class)
rl	Representation label

Returns

newly constructed gamma object

References GiNaC::clifford::clifford(), dirac_gamma(), dynallocate(), ex_to(), GiNaC::indexed::indexed(), is_a(), and symmetric2().

Referenced by dirac_gamma().

7.1.3.31 dirac_gamma5()

```
ex GiNaC::dirac_gamma5 (
             unsigned char rl = 0)
```

Create a Dirac gamma5 object.

Parameters

rl	Representation label

Returns

newly constructed object

References GiNaC::clifford::clifford(), dirac_gamma5(), and dynallocate().

Referenced by dirac_gamma5(), and dirac_trace().

7.1.3.32 dirac_gammaL()

```
ex GiNaC::dirac_gammaL (
             unsigned char rl = 0)
```

Create a Dirac gammaL object.

Parameters

rl	Representation label

Returns

newly constructed object

References GiNaC::clifford::clifford(), dirac_gammaL(), and dynallocate().

Referenced by dirac_gammaL(), and dirac_trace().

7.1.3.33 dirac_gammaR()

```
ex GiNaC::dirac_gammaR (
            unsigned char rl = 0)
```

Create a Dirac gammaR object.

Parameters

rl	Representation label

Returns

newly constructed object

References GiNaC::clifford::clifford(), dirac_gammaR(), and dynallocate().

Referenced by dirac_gammaR(), and dirac_trace().

7.1.3.34 dirac_slash()

```
ex GiNaC::dirac_slash (
            const ex & e,
            const ex & dim,
            unsigned char rl = 0)
```

Create a term of the form e_mu $*$ gamma$\sim$mu with a unique index mu.

Parameters

e	Original expression
dim	Dimension of index
rl	Representation label

References GiNaC::clifford::clifford(), dirac_slash(), dynallocate(), GiNaC::indexed::indexed(), symmetric2(), and GiNaC::varidx::varidx().

Referenced by GiNaC::diracgamma::contract_with(), and dirac_slash().

7.1.3.35 get_representation_label() [1/2]

```
unsigned char GiNaC::get_representation_label (
            const return_type_t & ti)  [static]
```

Extract representation label from tinfo key (as returned by return_type_tinfo()).

References get_representation_label(), and GiNaC::return_type_t::rl.

Referenced by canonicalize_clifford(), clifford_to_lst(), color_trace(), dirac_trace(), get_representation_label(), lst_to_clifford(), and remove_dirac_ONE().

7.1.3.36 trace_string()

```
ex GiNaC::trace_string (
            exvector::const_iterator ix,
            size_t num)  [static]
```

Take trace of a string of an even number of Dirac gammas given a vector of indices.

References lorentz_g(), and trace_string().

Referenced by dirac_trace(), and trace_string().

7.1.3.37 dirac_trace() [1/3]

```
ex GiNaC::dirac_trace (
            const ex & e,
            const std::set< unsigned char > & rls,
            const ex & trONE = 4)
```

Calculate dirac traces over the specified set of representation labels.

The computed trace is a linear functional that is equal to the usual trace only in D = 4 dimensions. In particular, the functional is not always cyclic in D != 4 dimensions when gamma5 is involved.

Parameters

e	Expression to take the trace of
rls	Set of representation labels
trONE	Expression to be returned as the trace of the unit matrix

References _ex0, _ex1, _ex4, base_and_index(), dirac_gamma5(), dirac_gammaL(), dirac_gammaR(), dirac_ONE(), dirac_trace(), ex_to(), expand(), get_representation_label(), I, is_a(), is_clifford_tinfo(), is_exactly_a(), lorentz_eps(), lorentz_g(), GiNaC::ex::map(), GiNaC::mul::mul(), GiNaC::subs_options::no_pattern, GiNaC::ex::nops(), GiNaC::ex::op(), permutation_sign(), GiNaC::ex::return_type_tinfo(), simplify_indexed(), GiNaC::ex::subs(), and trace_string().

Referenced by dirac_trace(), dirac_trace(), and dirac_trace().

7.1.3.38 dirac_trace() [2/3]

```
ex GiNaC::dirac_trace (
            const ex & e,
            const lst & rll,
            const ex & trONE = 4)
```

Calculate dirac traces over the specified list of representation labels.

The computed trace is a linear functional that is equal to the usual trace only in D = 4 dimensions. In particular, the functional is not always cyclic in D != 4 dimensions when gamma5 is involved.

Parameters

e	Expression to take the trace of
rll	List of representation labels
trONE	Expression to be returned as the trace of the unit matrix

References dirac_trace(), ex_to(), GiNaC::info_flags::nonnegint, and to_int().

7.1.3.39 dirac_trace() [3/3]

```
ex GiNaC::dirac_trace (
              const ex & e,
              unsigned char rl = 0,
              const ex & trONE = 4)
```

Calculate the trace of an expression containing gamma objects with a specified representation label.

The computed trace is a linear functional that is equal to the usual trace only in D = 4 dimensions. In particular, the functional is not always cyclic in D != 4 dimensions when gamma5 is involved.

Parameters

e	Expression to take the trace of
rl	Representation label
trONE	Expression to be returned as the trace of the unit matrix

References dirac_trace().

7.1.3.40 canonicalize_clifford()

```
ex GiNaC::canonicalize_clifford (
              const ex & e)
```

Bring all products of clifford objects in an expression into a canonical order.

This is not necessarily the most simple form but it will allow to check two expressions for equality.

References _ex2, base_and_index(), canonicalize_clifford(), dirac_ONE(), ex_to(), GiNaC::ex::expand(), get_representation_label(), GiNaC::ex::info(), is_a(), is_clifford_tinfo(), is_exactly_a(), lhs(), GiNaC::info_flags::list, GiNaC::ex::map(), GiNaC::ncmul::ncmul(), GiNaC::subs_options::no_pattern, GiNaC::return_types::noncommutative, GiNaC::ex::nops(), GiNaC::ex::op(), GiNaC::ex::return_type(), GiNaC::ex::return_type_tinfo(), rhs(), GiNaC::ex::simplify_indexed(), simplify_indexed(), GiNaC::ex::subs(), and GiNaC::ex::to_rational().

Referenced by canonicalize_clifford(), clifford_moebius_map(), clifford_to_lst(), and remove_dirac_ONE().

7.1.3.41 clifford_star_bar()

```
ex GiNaC::clifford_star_bar (
              const ex & e,
              bool do_bar,
              unsigned options)
```

An auxillary function performing clifford_star() and clifford_bar().

References clifford_star_bar(), dynallocate(), GiNaC::ex::expand(), GiNaC::ex::info(), is_a(), GiNaC::info_flags::list, GiNaC::ex::map(), GiNaC::ex::nops(), GiNaC::ex::op(), and pow().

Referenced by clifford_bar(), clifford_star(), and clifford_star_bar().

7.1.3.42 clifford_prime()

```
ex GiNaC::clifford_prime (
              const ex & e)
```

Automorphism of the Clifford algebra, simply changes signs of all clifford units.

References clifford_prime(), GiNaC::ex::info(), is_a(), GiNaC::info_flags::list, GiNaC::ex::map(), GiNaC::ex::op(), and pow().

Referenced by clifford_prime(), and clifford_to_lst().

7.1.3.43 remove_dirac_ONE()

```
ex GiNaC::remove_dirac_ONE (
              const ex & e,
              unsigned char rl = 0,
              unsigned options = 0)
```

Replaces dirac_ONE's (with a representation_label no less than rl) in e with 1.

For the default value rl = 0 remove all of them. Aborts if e contains any clifford_unit with representation_label to be removed.

Parameters

e	Expression to be processed
rl	Value of representation label
options	Defines some internal use

References canonicalize_clifford(), ex_to(), expand_dummy_sum(), get_representation_label(), GiNaC::ex::info(), is_a(), GiNaC::info_flags::list, GiNaC::ex::map(), GiNaC::ex::op(), pow(), and remove_dirac_ONE().

Referenced by clifford_norm(), clifford_to_lst(), and remove_dirac_ONE().

7.1.3.44 clifford_max_label()

```
int GiNaC::clifford_max_label (
              const ex & e,
              bool ignore_ONE = false)
```

Returns the maximal representation label of a clifford object if e contains at least one, otherwise returns -1.

Parameters

e	Expression to be processed
ignore_ONE	defines if clifford_ONE should be ignored in the search

References clifford_max_label(), ex_to(), is_a(), GiNaC::ex::nops(), and GiNaC::ex::op().

Referenced by clifford_max_label(), and GiNaC::add::coeff().

7.1.3.45 clifford_norm()

```
ex GiNaC::clifford_norm (
              const ex & e)
```

Calculation of the norm in the Clifford algebra.

References clifford_bar(), clifford_norm(), remove_dirac_ONE(), and sqrt().

Referenced by clifford_inverse(), and clifford_norm().

7.1.3.46 clifford_inverse()

```
ex GiNaC::clifford_inverse (
              const ex & e)
```

Calculation of the inverse in the Clifford algebra.

References clifford_bar(), clifford_inverse(), clifford_norm(), GiNaC::ex::is_zero(), and pow().

Referenced by clifford_inverse(), and clifford_moebius_map().

7.1.3.47 lst_to_clifford() [1/2]

```
ex GiNaC::lst_to_clifford (
              const ex & v,
              const ex & mu,
              const ex & metr,
              unsigned char rl = 0)
```

List or vector conversion into the Clifford vector.

Parameters

v	List or vector of coordinates
mu	Index (must be of class varidx or a derived class)
metr	Metric (should be indexed, tensmetric or a derived class, or a matrix)
rl	Representation label

Returns

Clifford vector with given components

References clifford_unit(), ex_to(), and lst_to_clifford().

Referenced by clifford_moebius_map(), lst_to_clifford(), and lst_to_clifford().

7.1.3.48 lst_to_clifford() [2/2]

```
ex GiNaC::lst_to_clifford (
              const ex & v,
              const ex & e)
```

List or vector conversion into the Clifford vector.

Parameters

v	List or vector of coordinates
e	Clifford unit object

Returns

Clifford vector with given components

References cols(), dirac_ONE(), ex_to(), get_dim_uint(), get_representation_label(), GiNaC::indexed::indexed(), GiNaC::ex::info(), is_a(), GiNaC::info_flags::list, lst_to_clifford(), GiNaC::matrix::matrix(), nops(), GiNaC::ex::op(), rows(), and sub_matrix().

7.1.3.49 get_clifford_comp()

```
ex GiNaC::get_clifford_comp (
              const ex & e,
              const ex & c,
              bool root = true)  [static]
```

Auxiliary structure to define a function for striping one Clifford unit from vectors.

Used in clifford_to_lst().

References dynallocate(), ex_to(), GiNaC::ex::expand(), get_all_dummy_indices(), get_clifford_comp(), GiNaC::ex::info(), is_a(), GiNaC::ex::is_zero(), GiNaC::info_flags::list, GiNaC::ex::map(), GiNaC::subs_options::no_pattern, GiNaC::ex::nops(), GiNaC::ex::op(), op(), and GiNaC::ex::subs().

Referenced by clifford_to_lst(), and get_clifford_comp().

7.1.3.50 clifford_to_lst()

```
lst GiNaC::clifford_to_lst (
              const ex & e,
              const ex & c,
              bool algebraic = true)
```

An inverse function to lst_to_clifford().

For given Clifford vector extracts its components with respect to given Clifford unit. Obtained components may contain Clifford units with a different metric. Extraction is based on the algebraic formula (e ∗ c.i + c.i ∗ e)/ pow(e.i, 2) for non-degenerate cases (i.e. neither pow(e.i, 2) = 0).

Parameters

e	Clifford expression to be decomposed into components
c	Clifford unit defining the metric for splitting (should have numeric dimension of indices)
algebraic	Use algebraic or symbolic algorithm for extractions

Returns

List of components of a Clifford vector

References GiNaC::container< C >::append(), canonicalize_clifford(), clifford_prime(), clifford_to_lst(), dirac_ONE(), ex_to(), expand_dummy_sum(), get_clifford_comp(), get_representation_label(), GINAC_ASSERT, is_a(), GiNaC::ex::is_zero(), GiNaC::numeric::is_zero(), GiNaC::subs_options::no_pattern, GiNaC::ex::op(), pow(), GiNaC::container< C >::remove_all(), remove_dirac_ONE(), simplify_indexed(), and GiNaC::ex::subs().

Referenced by clifford_moebius_map(), and clifford_to_lst().

7.1.3.51 clifford_moebius_map() [1/2]

```
ex GiNaC::clifford_moebius_map (
              const ex & a,
              const ex & b,
              const ex & c,
              const ex & d,
              const ex & v,
              const ex & G,
              unsigned char rl = 0)
```

Calculations of Moebius transformations (conformal map) defined by a 2x2 Clifford matrix (a b\c d) in linear spaces with arbitrary signature.

The expression is (a ∗ x + b)/(c ∗ x + d), where x is a vector build from list v with metric G. (see Jan Cnops. An introduction to {D}irac operators on manifolds, v.24 of Progress in Mathematical Physics. Birkhauser Boston Inc., Boston, MA, 2002.)

Parameters

a	(1,1) entry of the defining matrix
b	(1,2) entry of the defining matrix
c	(2,1) entry of the defining matrix
d	(2,2) entry of the defining matrix
v	Vector to be transformed
G	Metric of the surrounding space, may be a Clifford unit then the next parameter is ignored
rl	Representation label

Returns

List of components of the transformed vector

References canonicalize_clifford(), clifford_inverse(), clifford_moebius_map(), clifford_to_lst(), clifford_unit(), cols(), dynallocate(), ex_to(), G(), GiNaC::ex::info(), is_a(), GiNaC::info_flags::list, lst_to_clifford(), GiNaC::matrix::matrix(), GiNaC::container< C >::op(), rows(), and simplify_indexed().

Referenced by clifford_moebius_map(), and clifford_moebius_map().

7.1.3.52 clifford_moebius_map() [2/2]

```
ex GiNaC::clifford_moebius_map (
              const ex & M,
              const ex & v,
              const ex & G,
              unsigned char rl = 0)
```

The second form of Moebius transformations defined by a 2x2 Clifford matrix M This function takes the transformation matrix M as a single entity.

Parameters

M	the defining matrix
v	Vector to be transformed
G	Metric of the surrounding space, may be a Clifford unit then the next parameter is ignored
rl	Representation label

Returns

List of components of the transformed vector

References clifford_moebius_map(), cols(), ex_to(), G(), is_a(), GiNaC::ex::op(), and rows().

7.1.3.53 GINAC_DECLARE_UNARCHIVER() [2/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             clifford )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.54 GINAC_DECLARE_UNARCHIVER() [3/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             diracone )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.55 GINAC_DECLARE_UNARCHIVER() [4/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             cliffordunit )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.56 GINAC_DECLARE_UNARCHIVER() [5/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             diracgamma )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.57 GINAC_DECLARE_UNARCHIVER() [6/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             diracgamma5 )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.58 GINAC_DECLARE_UNARCHIVER() [7/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            diracgammaL )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.59 GINAC_DECLARE_UNARCHIVER() [8/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            diracgammaR )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.60 is_clifford_tinfo()

```
bool GiNaC::is_clifford_tinfo (
            const return_type_t & ti)  [inline]
```

Check whether a given return_type_t object (as returned by return_type_tinfo() is that of a clifford object (with an arbitrary representation label).

Parameters

ti	tinfo key

References is_clifford_tinfo(), and GiNaC::return_type_t::tinfo.

Referenced by canonicalize_clifford(), GiNaC::ncmul::conjugate(), dirac_trace(), and is_clifford_tinfo().

7.1.3.61 clifford_bar()

```
ex GiNaC::clifford_bar (
            const ex & e)  [inline]
```

Main anti-automorphism of the Clifford algebra: makes reversion and changes signs of all clifford units.

References clifford_bar(), and clifford_star_bar().

Referenced by clifford_bar(), clifford_inverse(), and clifford_norm().

7.1.3.62 clifford_star()

```
ex GiNaC::clifford_star (
            const ex & e)  [inline]
```

Reversion of the Clifford algebra, reverse the order of all clifford units in ncmul.

References clifford_star(), and clifford_star_bar().

Referenced by clifford_star().

7.1.3.63 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [4/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            su3one ,
            tensor ,
            print_func< print_dflt > &::do_print.  print_func< print_latex > &::do_print_↩
latex )
```

References GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.64 print_func< print_dflt >() [2/3]

```
GiNaC::print_func< print_dflt > (
            &su3t::do_print ) &
```

References GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.65 GINAC_BIND_UNARCHIVER() [9/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            color )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.66 GINAC_BIND_UNARCHIVER() [10/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            su3one )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.67 GINAC_BIND_UNARCHIVER() [11/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            su3t )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.68 GINAC_BIND_UNARCHIVER() [12/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            su3f )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.69 GINAC_BIND_UNARCHIVER() [13/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            su3d )
```

References GINAC_ASSERT, GINAC_BIND_UNARCHIVER, and is_a().

7.1.3.70 permute_free_index_to_front()

```
ex GiNaC::permute_free_index_to_front (
            const exvector & iv3,
            const exvector & iv2,
            int & sig)  [static]
```

Given a vector iv3 of three indices and a vector iv2 of two indices that is a subset of iv3, return the (free) index that is in iv3 but not in iv2 and the sign introduced by permuting that index to the front.

Parameters

iv3	Vector of 3 indices
iv2	Vector of 2 indices, must be a subset of iv3
sig	Returns sign introduced by index permutation

Returns

the free index (the one that is in iv3 but not in iv2)

References GINAC_ASSERT, permute_free_index_to_front(), and TEST_PERMUTATION.

Referenced by GiNaC::su3d::contract_with(), GiNaC::su3f::contract_with(), and permute_free_index_to_front().

7.1.3.71 color_ONE()

```
ex GiNaC::color_ONE (
            unsigned char rl = 0)
```

Create the su(3) unity element.

This is an indexed object, although it has no indices.

Parameters

rl	Representation label

Returns

newly constructed unity element

References GiNaC::color::color(), color_ONE(), and dynallocate().

Referenced by color_ONE(), GiNaC::su3t::contract_with(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.72 color_T()

```
ex GiNaC::color_T (
              const ex & a,
              unsigned char rl = 0)
```

Create an su(3) generator.

Parameters

a	Index
rl	Representation label

Returns

newly constructed unity generator

References GiNaC::color::color(), color_T(), dynallocate(), ex_to(), and is_a().

Referenced by color_T(), color_trace(), GiNaC::su3d::contract_with(), GiNaC::su3f::contract_with(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.73 color_f()

```
ex GiNaC::color_f (
              const ex & a,
              const ex & b,
              const ex & c)
```

Create an su(3) antisymmetric structure constant.

Parameters

a	First index
b	Second index
c	Third index

Returns

newly constructed structure constant

References antisymmetric3(), color_f(), dynallocate(), ex_to(), GiNaC::indexed::indexed(), and is_a().

Referenced by color_f(), color_h(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.74 color_d()

```
ex GiNaC::color_d (
              const ex & a,
              const ex & b,
              const ex & c)
```

Create an su(3) symmetric structure constant.

Parameters

a	First index
b	Second index
c	Third index

Returns

newly constructed structure constant

References color_d(), dynallocate(), ex_to(), GiNaC::indexed::indexed(), is_a(), and symmetric3().

Referenced by color_d(), color_h(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.75 color_h()

```
ex GiNaC::color_h (
              const ex & a,
              const ex & b,
              const ex & c)
```

This returns the linear combination d.a.b.c+I∗f.a.b.c.

References color_d(), color_f(), color_h(), and I.

Referenced by color_h(), color_trace(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.76 is_color_tinfo()

```
bool GiNaC::is_color_tinfo (
              const return_type_t & ti)  [static]
```

Check whether a given tinfo key (as returned by return_type_tinfo() is that of a color object (with an arbitrary representation label).

References is_color_tinfo(), and GiNaC::return_type_t::tinfo.

Referenced by color_trace(), and is_color_tinfo().

7.1.3.77 get_representation_label() [2/2]

```
unsigned char GiNaC::get_representation_label (
            const return_type_t & ti)  [static]
```

Extract representation label from tinfo key (as returned by return_type_tinfo()).

References get_representation_label(), and GiNaC::return_type_t::rl.

Referenced by GiNaC::su3d::contract_with(), and GiNaC::su3f::contract_with().

7.1.3.78 color_trace() [1/3]

```
ex GiNaC::color_trace (
            const ex & e,
            const std::set< unsigned char > & rls)
```

Calculate color traces over the specified set of representation labels.

Parameters

e	Expression to take the trace of
rls	Set of representation labels

References _ex0, _ex1, _ex3, color_h(), color_T(), color_trace(), delta_tensor(), dynallocate(), ex_to(), GiNaC::ex::expand(), get_representation_label(), is_a(), is_color_tinfo(), is_exactly_a(), GiNaC::ex::map(), GiNaC::ncmul::ncmul(), GiNaC::ex::nops(), GiNaC::ex::op(), and GiNaC::ex::return_type_tinfo().

Referenced by color_trace(), color_trace(), color_trace(), GiNaC::su3t::contract_with(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.79 color_trace() [2/3]

```
ex GiNaC::color_trace (
            const ex & e,
            const lst & rll)
```

Calculate color traces over the specified list of representation labels.

Parameters

e	Expression to take the trace of
rll	List of representation labels

References color_trace(), ex_to(), GiNaC::info_flags::nonnegint, and to_int().

7.1.3.80 color_trace() [3/3]

```
ex GiNaC::color_trace (
              const ex & e,
              unsigned char rl = 0)
```

Calculate the trace of an expression containing color objects with a specified representation label.

Parameters

e	Expression to take the trace of
rl	Representation label

References color_trace().

7.1.3.81 GINAC_DECLARE_UNARCHIVER() [9/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
              color )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.82 GINAC_DECLARE_UNARCHIVER() [10/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
              su3one )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.83 GINAC_DECLARE_UNARCHIVER() [11/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
              su3t )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.84 GINAC_DECLARE_UNARCHIVER() [12/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
              su3f )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.85 GINAC_DECLARE_UNARCHIVER() [13/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
              su3d )
```

References color_d(), color_f(), color_h(), color_ONE(), color_T(), color_trace(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.86 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [5/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            constant ,
            basic ,
            print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_tree > &::do_print_tree.  print_func< print_python_repr >
&::do_print_python_repr ) const
```

References GiNaC::constant::do_print(), GiNaC::constant::do_print_latex(), GiNaC::constant::do_print_python_repr(), GiNaC::constant::do_print_tree(), GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, and GINAC_IMPLEMENT_REGIS

7.1.3.87 GINAC_BIND_UNARCHIVER() [14/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            constant )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.88 GINAC_DECLARE_UNARCHIVER() [14/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            constant )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.89 crc32()

```
unsigned GiNaC::crc32 (
            const char * c,
            const unsigned len,
            const unsigned crcinit)  [static]
```

References crc32(), and crctab.

Referenced by crc32().

7.1.3.90 are_ex_trivially_equal()

```
bool GiNaC::are_ex_trivially_equal (
            const ex & e1,
            const ex & e2)  [inline]
```

Compare two objects of class quickly without doing a deep tree traversal.

Returns

"true" if they are equal "false" if equality cannot be established quickly (e1 and e2 may still be equal, in this case.

Referenced by GiNaC::add::conjugate(), GiNaC::container< std::vector >::conjugate(), GiNaC::expair::conjugate(), GiNaC::expairseq::conjugate(), GiNaC::integral::conjugate(), GiNaC::matrix::conjugate(), GiNaC::mul::conjugate(), GiNaC::power::conjugate(), GiNaC::pseries::conjugate(), GiNaC::pseries::eval_integ(), GiNaC::Gt::evalf(), GiNaC::integral::evalf(), GiNaC::pseries::evalm(), GiNaC::indexed::expand(), GiNaC::integral::expand(), GiNaC::power::expand(), GiNaC::expairseq::expandchildren(), GiNaC::mul::expandchildren(), GiNaC::ncmul::expandchildren(), GiNaC::expairseq::make_flat(), GiNaC::make_flat_inserter::make_flat_inserter(), GiNaC::basic::map(), GiNaC::idx::map(), GiNaC::power::map(), GiNaC::relational::map(), GiNaC::const_iterator::operator==(), GiNaC::internal::_iter_rep::operator==(), GiNaC::basic::subs(), GiNaC::clifford::subs(), GiNaC::idx::subs(), GiNaC::power::subs(), GiNaC::relational::subs(), GiNaC::container< C >::subschildren(), and GiNaC::expairseq::subschildren().

7.1.3.91 operator<<() [3/16]

```
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const exvector & e)
```

References get_print_context(), operator<<(), and GiNaC::print_dflt::print_dflt().

7.1.3.92 operator<<() [4/16]

```
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const exset & e)
```

References get_print_context(), operator<<(), and GiNaC::print_dflt::print_dflt().

7.1.3.93 operator<<() [5/16]

```
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const exmap & e)
```

References get_print_context(), operator<<(), and GiNaC::print_dflt::print_dflt().

7.1.3.94 nops() [1/2]

```
size_t GiNaC::nops (
            const ex & thisex)  [inline]
```

References GiNaC::ex::nops(), and nops().

Referenced by lst_to_clifford(), nops(), and nops().

7.1.3.95 expand() [1/2]

```
ex GiNaC::expand (
            const ex & thisex,
            unsigned options = 0)  [inline]
```

References GiNaC::ex::expand(), and expand().

Referenced by dirac_trace(), divide(), divide_in_z(), expand(), expand(), gcd(), gcd_pf_pow(), log_expand(), prem(), quo(), rem(), and sprem().

7.1.3.96 conjugate()

```
ex GiNaC::conjugate (
            const ex & thisex)  [inline]
```

References conjugate(), and GiNaC::ex::conjugate().

Referenced by conjugate(), and conjugate_expl_derivative().

7.1.3.97 real_part()

```
ex GiNaC::real_part (
            const ex & thisex)  [inline]
```

References GiNaC::ex::real_part(), and real_part().

Referenced by cos_imag_part(), cos_real_part(), cosh_imag_part(), cosh_real_part(), exp_imag_part(), exp_real_part(), log_imag_part(), real_part(), real_part_expl_derivative(), sin_imag_part(), sin_real_part(), sinh_imag_part(), sinh_real_part(), tan_imag_part(), tan_real_part(), tanh_imag_part(), and tanh_real_part().

7.1.3.98 imag_part()

```
ex GiNaC::imag_part (
            const ex & thisex)  [inline]
```

References GiNaC::ex::imag_part(), and imag_part().

Referenced by cos_imag_part(), cos_real_part(), cosh_imag_part(), cosh_real_part(), exp_imag_part(), exp_real_part(), imag_part(), imag_part_expl_derivative(), log_imag_part(), sin_imag_part(), sin_real_part(), sinh_imag_part(), sinh_real_part(), tan_imag_part(), tan_real_part(), tanh_imag_part(), and tanh_real_part().

7.1.3.99 has()

```
bool GiNaC::has (
            const ex & thisex,
            const ex & pattern,
            unsigned options = 0)  [inline]
```

References GiNaC::ex::has(), and has().

Referenced by has().

7.1.3.100 find()

```
bool GiNaC::find (
            const ex & thisex,
            const ex & pattern,
            exset & found)  [inline]
```

References GiNaC::ex::find(), and find().

Referenced by find().

7.1.3.101 is_polynomial()

```
bool GiNaC::is_polynomial (
            const ex & thisex,
            const ex & vars)  [inline]
```

References GiNaC::ex::is_polynomial(), and is_polynomial().

Referenced by is_polynomial().

7.1.3.102 degree()

```
int GiNaC::degree (
            const ex & thisex,
            const ex & s)  [inline]
```

References degree(), and GiNaC::ex::degree().

Referenced by degree(), replace_with_symbol(), and sqrfree_parfrac().

7.1.3.103 ldegree()

```
int GiNaC::ldegree (
            const ex & thisex,
            const ex & s)  [inline]
```

References GiNaC::ex::ldegree(), and ldegree().

Referenced by ldegree().

7.1.3.104 coeff()

```
ex GiNaC::coeff (
            const ex & thisex,
            const ex & s,
            int n = 1)  [inline]
```

References coeff(), and GiNaC::ex::coeff().

Referenced by coeff(), GiNaC::make_flat_inserter::handle_factor(), iterated_integral_evalf_impl(), log_series(), and sqrfree_parfrac().

7.1.3.105 numer() [1/2]

```
ex GiNaC::numer (
            const ex & thisex)  [inline]
```

References GiNaC::ex::numer(), and numer().

Referenced by decomp_rational(), GiNaC::power::imag_part(), GiNaC::add::integer_content(), numer(), numer(), print_real_csrc(), GiNaC::power::real_part(), replace_with_symbol(), and sqrfree_parfrac().

7.1.3.106 denom() [1/2]

```
ex GiNaC::denom (
            const ex & thisex)  [inline]
```

References denom(), and GiNaC::ex::denom().

Referenced by decomp_rational(), denom(), denom(), GiNaC::add::integer_content(), lcmcoeff(), print_real_csrc(), replace_with_symbol(), and sqrfree_parfrac().

7.1.3.107 numer_denom()

```
ex GiNaC::numer_denom (
             const ex & thisex)  [inline]
```

References GiNaC::ex::numer_denom(), and numer_denom().

Referenced by decomp_rational(), numer_denom(), and sqrfree_parfrac().

7.1.3.108 normal()

```
ex GiNaC::normal (
             const ex & thisex)  [inline]
```

References GiNaC::ex::normal(), and normal().

Referenced by fsolve(), normal(), GiNaC::normal_map_function::operator()(), and replace_with_symbol().

7.1.3.109 to_rational()

```
ex GiNaC::to_rational (
             const ex & thisex,
             exmap & repl)  [inline]
```

References GiNaC::ex::to_rational(), and to_rational().

Referenced by to_rational().

7.1.3.110 to_polynomial()

```
ex GiNaC::to_polynomial (
             const ex & thisex,
             exmap & repl)  [inline]
```

References GiNaC::ex::to_polynomial(), and to_polynomial().

Referenced by to_polynomial().

7.1.3.111 collect()

```
ex GiNaC::collect (
             const ex & thisex,
             const ex & s,
             bool distributed = false)  [inline]
```

References collect(), and GiNaC::ex::collect().

Referenced by collect().

7.1.3.112 eval()

```
ex GiNaC::eval (
            const ex & thisex)  [inline]
```

References eval(), and GiNaC::ex::eval().

Referenced by eval().

7.1.3.113 evalf() [1/2]

```
ex GiNaC::evalf (
            const ex & thisex)  [inline]
```

References evalf(), and GiNaC::ex::evalf().

Referenced by beta_eval(), eta_evalf(), evalf(), evalf(), GiNaC::evalf_map_function::operator()(), and zeta2_evalf().

7.1.3.114 evalm()

```
ex GiNaC::evalm (
            const ex & thisex)  [inline]
```

References evalm(), and GiNaC::ex::evalm().

Referenced by evalm(), and GiNaC::evalm_map_function::operator()().

7.1.3.115 eval_integ()

```
ex GiNaC::eval_integ (
            const ex & thisex)  [inline]
```

References eval_integ(), and GiNaC::ex::eval_integ().

Referenced by eval_integ(), and GiNaC::eval_integ_map_function::operator()().

7.1.3.116 diff()

```
ex GiNaC::diff (
            const ex & thisex,
            const symbol & s,
            unsigned nth = 1)  [inline]
```

References diff(), and GiNaC::ex::diff().

Referenced by diff(), and GiNaC::derivative_map_function::operator()().

7.1.3.117 series()

```
ex GiNaC::series (
             const ex & thisex,
             const ex & r,
             int order,
             unsigned options = 0)  [inline]
```

References GiNaC::ex::series(), and series().

Referenced by atan_series(), atanh_series(), Li2_series(), log_series(), and series().

7.1.3.118 match()

```
bool GiNaC::match (
             const ex & thisex,
             const ex & pattern,
             exmap & repl_lst)  [inline]
```

References GiNaC::ex::match(), and match().

Referenced by match().

7.1.3.119 simplify_indexed() [1/3]

```
ex GiNaC::simplify_indexed (
             const ex & thisex,
             unsigned options = 0)  [inline]
```

References GiNaC::ex::simplify_indexed(), and simplify_indexed().

Referenced by canonicalize_clifford(), clifford_moebius_map(), clifford_to_lst(), dirac_trace(), GiNaC::ex::simplify_indexed(), GiNaC::ex::simplify_indexed(), simplify_indexed(), and simplify_indexed().

7.1.3.120 simplify_indexed() [2/3]

```
ex GiNaC::simplify_indexed (
             const ex & thisex,
             const scalar_products & sp,
             unsigned options = 0)  [inline]
```

References GiNaC::ex::simplify_indexed(), and simplify_indexed().

7.1.3.121 symmetrize() [1/4]

```
ex GiNaC::symmetrize (
             const ex & thisex)  [inline]
```

References GiNaC::ex::symmetrize(), and symmetrize().

Referenced by idx_symmetrization(), GiNaC::ex::symmetrize(), symmetrize(), symmetrize(), symmetrize(), symmetrize(), and symmetrize_cyclic().

7.1.3.122 symmetrize() [2/4]

```
ex GiNaC::symmetrize (
             const ex & thisex,
             const lst & l)  [inline]
```

References GiNaC::ex::symmetrize(), and symmetrize().

7.1.3.123 antisymmetrize() [1/4]

```
ex GiNaC::antisymmetrize (
             const ex & thisex)  [inline]
```

References antisymmetrize(), and GiNaC::ex::antisymmetrize().

Referenced by antisymmetrize(), antisymmetrize(), antisymmetrize(), antisymmetrize(), and GiNaC::ex::antisymmetrize().

7.1.3.124 antisymmetrize() [2/4]

```
ex GiNaC::antisymmetrize (
             const ex & thisex,
             const lst & l)  [inline]
```

References antisymmetrize(), and GiNaC::ex::antisymmetrize().

7.1.3.125 symmetrize_cyclic() [1/4]

```
ex GiNaC::symmetrize_cyclic (
             const ex & thisex)  [inline]
```

References GiNaC::ex::symmetrize_cyclic(), and symmetrize_cyclic().

Referenced by GiNaC::ex::symmetrize_cyclic(), GiNaC::ex::symmetrize_cyclic(), symmetrize_cyclic(), symmetrize_cyclic(), symmetrize_cyclic(), and symmetrize_cyclic().

7.1.3.126 symmetrize_cyclic() [2/4]

```
ex GiNaC::symmetrize_cyclic (
             const ex & thisex,
             const lst & l)  [inline]
```

References GiNaC::ex::symmetrize_cyclic(), and symmetrize_cyclic().

7.1.3.127 op()

```
ex GiNaC::op (
             const ex & thisex,
             size_t i)  [inline]
```

References GiNaC::ex::op(), and op().

Referenced by get_clifford_comp(), Li2_series(), op(), and rename_dummy_indices().

7.1.3.128 lhs()

```
ex GiNaC::lhs (
            const ex & thisex)  [inline]
```

References GiNaC::ex::lhs(), and lhs().

Referenced by canonicalize_clifford(), lhs(), GiNaC::ptr< T >::operator!=, GiNaC::ptr< T >::operator!=, GiNaC::const_iterator::operator-, GiNaC::ptr< T >::operator==, and GiNaC::ptr< T >::operator==.

7.1.3.129 rhs()

```
ex GiNaC::rhs (
            const ex & thisex)  [inline]
```

References GiNaC::ex::rhs(), and rhs().

Referenced by canonicalize_clifford(), lsolve(), GiNaC::ptr< T >::operator!=, GiNaC::ptr< T >::operator!=(), GiNaC::ptr< T >::operator!=, GiNaC::const_iterator::operator-, GiNaC::ptr< T >::operator<<, GiNaC::ptr< T >::operator==, GiNaC::ptr< T >::operator==(), GiNaC::ptr< T >::operator==, rhs(), GiNaC::matrix::solve(), and sqrfree_parfrac().

7.1.3.130 is_zero() [1/2]

```
bool GiNaC::is_zero (
            const ex & thisex)  [inline]
```

References GiNaC::ex::is_zero(), and is_zero().

Referenced by cosh_eval(), GiNaC::matrix::determinant(), GiNaC::matrix::determinant_minor(), GiNaC::tensepsilon::eval_indexed(), GiNaC::matrix::fraction_free_elimination(), GiNaC::matrix::gauss_elimination(), is_zero(), is_zero(), GiNaC::matrix::markowitz_elimina GiNaC::matrix::mul(), GiNaC::relational::operator safe_bool(), GiNaC::Gt_detail::pathintegral_term::pathintegral_term(), GiNaC::matrix::pivot(), GiNaC::Gt::regularise(), sinh_eval(), tanh_eval(), GiNaC::Gt::tauToFundamental(), and GiNaC::Gt::zisToFundamental().

7.1.3.131 swap() [1/2]

```
void GiNaC::swap (
            ex & e1,
            ex & e2)  [inline]
```

References GiNaC::ex::swap(), and swap().

Referenced by permutation_sign(), GiNaC::matrix::pivot(), swap(), and swap().

7.1.3.132 subs() [1/3]

```
ex GiNaC::subs (
            const ex & thisex,
            const exmap & m,
            unsigned options = 0)  [inline]
```

References GiNaC::ex::subs(), and subs().

Referenced by Li2_series(), subs(), subs(), and subs().

7.1.3.133 subs() [2/3]

```
ex GiNaC::subs (
             const ex & thisex,
             const lst & ls,
             const lst & lr,
             unsigned options = 0)  [inline]
```

References GiNaC::ex::subs(), and subs().

7.1.3.134 subs() [3/3]

```
ex GiNaC::subs (
             const ex & thisex,
             const ex & e,
             unsigned options = 0)  [inline]
```

References GiNaC::ex::subs(), and subs().

7.1.3.135 is_a() [2/3]

```
template<class T>
bool GiNaC::is_a (
             const ex & obj)  [inline]
```

Check if ex is a handle to a T, including base classes.

7.1.3.136 is_exactly_a() [2/2]

```
template<class T>
bool GiNaC::is_exactly_a (
             const ex & obj)  [inline]
```

Check if ex is a handle to a T, not including base classes.

7.1.3.137 ex_to()

```
template<class T>
attribute_pure const T & GiNaC::ex_to (
             const ex & e)  [inline]
```

Return a reference to the basic-derived class T object embedded in an expression.

This is fast but unsafe: the result is undefined if the expression does not contain a T object at its top level. Hence, you should generally check the type of e first. Also, you shouldn't cache the returned reference because GiNaC's garbage collector may destroy the referenced object any time it's used in another expression.

Parameters

e	expression

Returns

reference to object of class T

See also

is_exactly_a<class T>()

Referenced by abs_eval(), abs_evalf(), abs_power(), acos_eval(), acos_evalf(), acosh_eval(), acosh_evalf(), adaptivesimpson(), GiNaC::symmetry::add(), GiNaC::matrix::add_indexed(), GiNaC::pseries::add_series(), antisymmetric2(), antisymmetric3(), antisymmetric4(), GiNaC::Gt::apply_function_recursive(), asin_eval(), asin_evalf(), asinh_conjugate(), asinh_eval(), asinh_evalf(), atan2_eval(), atan2_evalf(), atan_conjugate(), atan_eval(), atan_evalf(), atan_series(), atanh_eval(), atanh_evalf(), atanh_series(), base_and_index(), beta_eval(), beta_evalf(), beta_series(), binomial_eval(), GiNaC::mul::can_be_further_expanded(), GiNaC::symmetry::canonicalize, canonicalize_clifford(), clifford_max_label(), clifford_moebius_map(), clifford_moebius_map(), clifford_to_lst(), clifford_unit(), GiNaC::basic::coeff(), GiNaC::ncmul::coeff(), GiNaC::power::coeff(), GiNaC::pseries::coeff(), GiNaC::basic::collect(), color_d(), color_f(), color_T(), color_trace(), color_trace(), GiNaC::add::combine_ex_with_coeff_to_pair(), GiNaC::expairseq::combine_overall_coeff(), GiNaC::expairseq::combine_overall_coeff(), GiNaC::mul::combine_overall_coeff(), GiNaC::mul::combine_overall_coeff(), GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::expairseq::combine_pair_with_coeff_ GiNaC::expairseq::combine_same_terms_sorted_seq(), conjugate_evalf(), conjugate_expl_derivative(), GiNaC::expairseq::construct_ GiNaC::expairseq::construct_from_2_expairseq(), GiNaC::expairseq::construct_from_expairseq_ex(), GiNaC::cliffordunit::contract_wi GiNaC::diracgamma::contract_with(), GiNaC::matrix::contract_with(), GiNaC::spinmetric::contract_with(), GiNaC::su3d::contract_with GiNaC::su3f::contract_with(), GiNaC::su3t::contract_with(), cos_eval(), cos_evalf(), cosh_eval(), cosh_evalf(), csgn_eval(), csgn_evalf(), csgn_power(), csgn_series(), GiNaC::basic::degree(), GiNaC::mul::degree(), GiNaC::ncmul::degree(), GiNaC::power::degree(), GiNaC::pseries::degree(), dirac_gamma(), dirac_trace(), dirac_trace(), divide(), divide_in_z(), GiNaC::add::do_print_csrc(), GiNaC::idx::do_print_csrc(), GiNaC::mul::do_print_csrc(), GiNaC::power::do_print_csrc(), GiNaC::integral::do_print_latex(), GiNaC::mul::do_print_latex(), GiNaC::power::do_print_latex(), EllipticE_evalf(), EllipticK_evalf(), epsilon_tensor(), epsilon_tensor(), eta_eval(), eta_evalf(), GiNaC::function::eval(), GiNaC::indexed::eval(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::matrix::eval_indexed(), GiNaC::minkmetric::eval_indexed(), GiNaC::spinmetric::eval_indexed(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::tensdelta::eval_indexed(), GiNaC::tensepsilon::eval_indexed(), GiNaC::tensmetric::eval_indexed(), GiNaC::clifford::eval_ncmul(), GiNaC::add::evalm(), GiNaC::mul::evalm(), GiNaC::ncmul::evalm(), GiNaC::power::evalm(), exp_eval(), exp_evalf(), GiNaC::mul::expair_needs_further_proc GiNaC::basic::expand(), GiNaC::indexed::expand(), GiNaC::mul::expand(), GiNaC::power::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), expand_dummy_sum(), GiNaC::power::expand_mul(), factorial_eval(), find_free_and_dummy(), frac_cancel(), fsolve(), G2_eval(), G2_evalf(), G3_eval(), G3_evalf(), gcd(), gcd_pf_pow(), generalised_Bernoulli_number(), get_all_dummy_indices(), get_all_dummy_indices_safely(), get_clifford_comp(), get_dim_uint(), GiNaC::clifford::get_metric(), GiNaC::Kronecker_dtau_kernel::get_numerical_value(), GiNaC::Kronecker_dz_kernel::get_numerical_value(), GiNaC::integration_kernel::get_numerical_value_impl(), GiNaC::Gt::Gt(), GiNaC::Gt::Gt(), H_deriv(), H_eval(), H_evalf(), H_print_latex(), GiNaC::numeric::has(), GiNaC::power::has(), GiNaC::symmetry::has_cyclic(), GiNaC::symmetry::has_nonsymmetric(), heur_gcd_z(), GiNaC::power::imag_part(), imag_part_evalf(), imag_part_expl_derivative(), index0(), index1(), index2(), index3(), GiNaC::add::integer_content(), GiNaC::mul::integer_content(), is_discriminant_of_quadratic_nu is_dummy_pair(), GiNaC::basic::is_polynomial(), is_the_function(), iterated_integral_evalf_impl(), kronecker_symbol(), lcm(), lcmcoeff(), GiNaC::basic::ldegree(), GiNaC::mul::ldegree(), GiNaC::ncmul::ldegree(), GiNaC::power::ldegree(), GiNaC::pseries::ldegree(), lgamma_eval(), lgamma_evalf(), lgamma_series(), Li2_eval(), Li2_evalf(), Li2_series(), Li_eval(), Li_evalf(), Li_print_latex(), log_eval(), log_evalf(), log_expand(), log_series(), lorentz_eps(), lsolve(), GiNaC::Gt::lst_evaluate(), lst_to_clifford(), lst_to_clifford(), lst_to_matrix(), GiNaC::expairseq::make_flat(), GiNaC::expairseq::make_flat(), GiNaC::basic::match(), GiNaC::expairseq::match(), GiNaC::Gt::match(), GiNaC::wildcard::match(), GiNaC::add::max_coefficient(), GiNaC::mul::max_coefficient(), GiNaC::pseries::mul_series(), multiply_lcm(), GiNaC::add::normal(), GiNaC::mul::normal(), GiNaC::power::normal(), not_symmetric(), GiNaC::relational::operator safe_bool(), GiNaC::idx_is_equal_ignore_dim::operator()(), GiNaC::sy_is_less::operator()(), GiNaC::sy_swap::operator()(), GiNaC::basic::operator[](), GiNaC::basic::operator[](), Order_eval(), Order_series(), GiNaC::matrix::pivot(), GiNaC::matrix::pow(), GiNaC::add::print_add(), GiNaC::mul::print_overall_coeff(), GiNaC::indexed::printindices(),

psi1_eval(), psi1_evalf(), psi1_series(), psi2_eval(), psi2_evalf(), psi2_series(), GiNaC::Eisenstein_h_kernel::q_expansion_modular_fo
GiNaC::Eisenstein_kernel::q_expansion_modular_form(), GiNaC::Gt::qExpand(), GiNaC::indexed::read_archive(),
GiNaC::symbol::read_archive(), GiNaC::symmetry::read_archive(), GiNaC::power::real_part(), real_part_evalf(),
real_part_expl_derivative(), GiNaC::add::recombine_pair_to_ex(), remove_dirac_ONE(), rename_dummy_indices_uniquely(),
rename_dummy_indices_uniquely(), rename_dummy_indices_uniquely(), rename_dummy_indices_uniquely(),
GiNaC::tensor::replace_contr_index(), replace_with_symbol(), GiNaC::indexed::reposition_dummy_indices,
S_eval(), S_evalf(), GiNaC::clifford::same_metric(), GiNaC::matrix::scalar_mul_indexed(), GiNaC::add::series(),
GiNaC::basic::series(), GiNaC::integral::series(), GiNaC::mul::series(), GiNaC::power::series(), GiNaC::pseries::series(),
GiNaC::symbol::series(), GiNaC::integration_kernel::series_coeff(), GiNaC::Ebar_kernel::series_coeff_impl(),
GiNaC::ELi_kernel::series_coeff_impl(), GiNaC::Kronecker_dtau_kernel::series_coeff_impl(), GiNaC::Kronecker_dz_kernel::series_c
GiNaC::multiple_polylog_kernel::series_coeff_impl(), series_to_poly(), GiNaC::Gt::setArgs(), GiNaC::indexed::simplify_indexed,
GiNaC::indexed::simplify_indexed_product, sin_eval(), sin_evalf(), sinh_eval(), sinh_evalf(), GiNaC::add::smod(),
GiNaC::mul::smod(), spinor_metric(), GiNaC::add::split_ex_to_pair(), GiNaC::mul::split_ex_to_pair(), sqrfree(),
sqrfree_parfrac(), step_eval(), step_evalf(), step_series(), GiNaC::clifford::subs(), GiNaC::container< C >::subs(),
GiNaC::expairseq::subs(), GiNaC::power::subs(), GiNaC::basic::subs_one_level(), symmetric2(), symmetric3(),
symmetric4(), tan_eval(), tan_evalf(), tanh_eval(), tanh_evalf(), tgamma_eval(), tgamma_evalf(), tgamma_series(),
GiNaC::indexed::thiscontainer(), GiNaC::indexed::thiscontainer(), GiNaC::Gt::to_integer(), GiNaC::Gt::to_numeric(),
tryfactsubs(), GiNaC::indexed::validate(), zeta1_eval(), zeta1_evalf(), zeta1_print_latex(), zeta2_eval(), zeta2_evalf(),
and zeta2_print_latex().

7.1.3.138 compile_ex() [1/3]

```
void GiNaC::compile_ex (
            const ex & expr,
            const symbol & sym,
            FUNCP_1P & fp,
            const std::string filename = "")
```

Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.

The function pointer has type FUNCP_1P.

Parameters

expr	Expression to be compiled
sym	Symbol from the expression to become the function parameter
fp	Returned function pointer
filename	Name of the intermediate source code and so-file. If supplied, these intermediate files will not be deleted

References compile_ex().

Referenced by compile_ex(), compile_ex(), and compile_ex().

7.1.3.139 compile_ex() [2/3]

```
void GiNaC::compile_ex (
            const ex & expr,
            const symbol & sym1,
            const symbol & sym2,
```

```
            FUNCP_2P & fp,
            const std::string filename = "")
```

Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.

The function pointer has type FUNCP_2P.

Parameters

expr	Expression to be compiled
sym1	Symbol from the expression to become the first function parameter
sym2	Symbol from the expression to become the second function parameter
fp	Returned function pointer
filename	Name of the intermediate source code and so-file. If supplied, these intermediate files will not be deleted

References compile_ex().

7.1.3.140 compile_ex() [3/3]

```
void GiNaC::compile_ex (
            const lst & exprs,
            const lst & syms,
            FUNCP_CUBA & fp,
            const std::string filename = "")
```

Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.

The function pointer has type FUNCP_CUBA.

Parameters

exprs	List of expression to be compiled
syms	Symbols from the expression to become the function parameters
fp	Returned function pointer
filename	Name of the intermediate source code and so-file. If supplied, these intermediate files will not be deleted

References compile_ex().

7.1.3.141 link_ex() [1/2]

```
void GiNaC::link_ex (
            const std::string filename,
            FUNCP_1P & fp)
```

Opens an existing so-file and returns a function pointer of type FUNCP_1P to the contained function.

Opens an existing so-file and returns a function pointer of type FUNCP_2P to the contained function.

The so-file has to be generated by compile_ex in advance.

Parameters

filename	Name of the so-file to open and link
fp	Returned function pointer

References link_ex().

Referenced by link_ex(), and link_ex().

7.1.3.142 link_ex() [2/2]

```
void GiNaC::link_ex (
            const std::string filename,
            FUNCP_CUBA & fp)
```

Opens an existing so-file and returns a function pointer of type FUNCP_CUBA to the contained function.

The so-file has to be generated by compile_ex in advance.

Parameters

filename	Name of the so-file to open and link
fp	Returned function pointer

References link_ex().

7.1.3.143 unlink_ex()

```
void GiNaC::unlink_ex (
            const std::string filename)
```

Closes all linked .so files that have the supplied filename.

Parameters

filename	Name of the so-file to close

References unlink_ex().

Referenced by unlink_ex().

7.1.3.144 swap() [2/2]

```
void GiNaC::swap (
            expair & e1,
            expair & e2)  [inline]
```

References GiNaC::expair::swap(), and swap().

7.1.3.145 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [6/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             expairseq ,
             basic ,
             print_func< print_context > &::do_print.  print_func< print_tree > &::do_print↩
_tree )
```

References GiNaC::expairseq::do_print(), GiNaC::expairseq::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OP

7.1.3.146 conjugateepvector()

```
epvector * GiNaC::conjugateepvector (
             const epvector & )
```

Complex conjugate every element of an epvector.

Returns zero if this does not change anything.

References GiNaC::expair::conjugate(), conjugateepvector(), and GiNaC::expair::is_equal().

Referenced by GiNaC::expairseq::conjugate(), GiNaC::pseries::conjugate(), and conjugateepvector().

7.1.3.147 factor()

```
ex GiNaC::factor (
             const ex & poly,
             unsigned options)
```

Interface function to the outside world.

Factorizes univariate and multivariate polynomials.

It uses factor1() on each of the explicitly present factors of poly.

The default option is factor_options::polynomial, which means that factor() will only factorize an expression if it is a proper polynomial (i.e. the flag info_flags::polynomial is set). Given the option factor_options::all, factor() will factorize all subexpressions, e.g. polynomials containing functions or polynomials inside function arguments.

Parameters

in	*poly*	expression to factorize
in	*options*	see GiNaC::factor_options

Returns

factorized expression

References factor(), and pow().

Referenced by algebraic_match_mul_with_mul(), collect_common_factors(), GiNaC::ncmul::eval(), GiNaC::power::expand_add(), GiNaC::mul::expandchildren(), factor(), find_common_factor(), GiNaC::mul::find_real_imag(), GiNaC::mul::info(), GiNaC::Gt::qExpand(), GiNaC::mul::series(), sqrfree_parfrac(), and GiNaC::Gt::zisToFundamental().

7.1.3.148 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [7/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            fail ,
            basic ,
            print_func< print_context > &::do_print.  print_func< print_tree > &::do_print↩
_tree )
```

References DEFAULT_COMPARE, DEFAULT_CTOR, DEFAULT_PRINT, GiNaC::fail::do_print(), GiNaC::basic::do_print_tree(), GINAC_BIND_UNARCHIVER, and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.149 GINAC_DECLARE_UNARCHIVER() [15/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            fail )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.150 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [8/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            fderivative ,
            function ,
            print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_csrc > &::do_print_csrc.  print_func< print_tree > &::do_↩
print_tree )
```

References GiNaC::fderivative::do_print(), GiNaC::fderivative::do_print_csrc(), GiNaC::fderivative::do_print_latex(), GiNaC::fderivative::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.151 GINAC_BIND_UNARCHIVER() [15/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            fderivative )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.152 GINAC_DECLARE_UNARCHIVER() [16/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            fderivative )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.153 GINAC_BIND_UNARCHIVER() [16/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            function )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.154 GINAC_DECLARE_UNARCHIVER() [17/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            function )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.155 is_the_function()

```
template<typename T>
bool GiNaC::is_the_function (
            const ex & x)  [inline]
```

References ex_to(), is_exactly_a(), and is_the_function().

Referenced by is_the_function(), is_the_function< G_SERIAL >(), is_the_function< iterated_integral_SERIAL >(), is_the_function< psi_SERIAL >(), is_the_function< zeta_SERIAL >(), and GiNaC::Gt_detail::TransformExpressionWithCache< GiN

7.1.3.156 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [9/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            Gt ,
            basic ,
            print_func< print_dflt > &::do_print.  print_func< print_python > &::do_print.
print_func< print_latex > &::do_print_latex )
```

References GiNaC::Gt::do_print(), GiNaC::Gt::do_print_latex(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.157 GINAC_BIND_UNARCHIVER() [17/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            Gt )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.158 GINAC_DECLARE_UNARCHIVER() [18/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            Gt )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.159 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [10/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            idx ,
            basic ,
            print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_csrc > &::do_print_csrc.  print_func< print_tree > &::do_↩
print_tree )
```

References GiNaC::idx::do_print(), GiNaC::idx::do_print_csrc(), GiNaC::idx::do_print_latex(), GiNaC::idx::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.160 print_func< print_context >()

```
GiNaC::print_func< print_context > (
            &varidx::do_print ) &
```

7.1.3.161 GINAC_BIND_UNARCHIVER() [18/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            idx )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.162 GINAC_BIND_UNARCHIVER() [19/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            varidx )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.163 GINAC_BIND_UNARCHIVER() [20/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            spinidx )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.164 is_dummy_pair() [1/2]

```
bool GiNaC::is_dummy_pair (
            const idx & i1,
            const idx & i2)
```

Check whether two indices form a dummy pair.

References is_dummy_pair(), and GiNaC::idx::is_dummy_pair_same_type().

Referenced by GiNaC::matrix::contract_with(), GiNaC::spinmetric::contract_with(), GiNaC::matrix::eval_indexed(), GiNaC::tensdelta::eval_indexed(), find_free_and_dummy(), GINAC_DECLARE_UNARCHIVER(), GiNaC::indexed::has_dummy_index(), is_dummy_pair(), is_dummy_pair(), GiNaC::is_summation_idx::operator()(), GiNaC::tensor::replace_contr_index(), and GiNaC::indexed::reposition_dummy_indices.

7.1.3.165 is_dummy_pair() [2/2]

```
bool GiNaC::is_dummy_pair (
            const ex & e1,
            const ex & e2)
```

Check whether two expressions form a dummy index pair.

References ex_to(), is_a(), and is_dummy_pair().

7.1.3.166 find_free_and_dummy() [1/2]

```
void GiNaC::find_free_and_dummy (
             exvector::const_iterator it,
             exvector::const_iterator itend,
             exvector & out_free,
             exvector & out_dummy)
```

Given a vector of indices, split them into two vectors, one containing the free indices, the other containing the dummy indices (numeric indices are neither free nor dummy ones).

Parameters

it	Pointer to start of index vector
itend	Pointer to end of index vector
out_free	Vector of free indices (returned, sorted)
out_dummy	Vector of dummy indices (returned, sorted)

References ex_to(), find_free_and_dummy(), is_dummy_pair(), and shaker_sort().

Referenced by GiNaC::su3d::contract_with(), count_dummy_indices(), count_free_indices(), find_dummy_indices(), find_free_and_dummy(), find_free_and_dummy(), get_all_dummy_indices_safely(), GiNaC::indexed::get_dummy_indices(), GiNaC::indexed::get_free_indices(), GiNaC::mul::get_free_indices(), GiNaC::ncmul::get_free_indices(), GINAC_DECLARE_UNARCH GiNaC::indexed::simplify_indexed, and GiNaC::indexed::simplify_indexed_product.

7.1.3.167 minimal_dim()

```
ex GiNaC::minimal_dim (
             const ex & dim1,
             const ex & dim2)
```

Return the minimum of two index dimensions.

If this is undecidable, throw an exception. Numeric dimensions are always considered "smaller" than symbolic dimensions.

References is_a(), GiNaC::ex::is_equal(), is_exactly_a(), and minimal_dim().

Referenced by GiNaC::diracgamma::contract_with(), GiNaC::idx::minimal_dim(), minimal_dim(), and GiNaC::indexed::simplify_indexe

7.1.3.168 GINAC_DECLARE_UNARCHIVER() [19/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             idx )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.169 GINAC_DECLARE_UNARCHIVER() [20/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             varidx )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.170 GINAC_DECLARE_UNARCHIVER() [21/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            spinidx )
```

References find_free_and_dummy(), GINAC_DECLARE_UNARCHIVER(), and is_dummy_pair().

7.1.3.171 find_free_and_dummy() [2/2]

```
void GiNaC::find_free_and_dummy (
            const exvector & v,
            exvector & out_free,
            exvector & out_dummy)  [inline]
```

Given a vector of indices, split them into two vectors, one containing the free indices, the other containing the dummy indices (numeric indices are neither free nor dummy ones).

Parameters

v	Index vector
out_free	Vector of free indices (returned, sorted)
out_dummy	Vector of dummy indices (returned, sorted)

References find_free_and_dummy().

7.1.3.172 find_dummy_indices()

```
void GiNaC::find_dummy_indices (
            const exvector & v,
            exvector & out_dummy)  [inline]
```

Given a vector of indices, find the dummy indices.

Parameters

v	Index vector
out_dummy	Vector of dummy indices (returned, sorted)

References find_dummy_indices(), and find_free_and_dummy().

Referenced by find_dummy_indices(), and GiNaC::indexed::get_dummy_indices().

7.1.3.173 count_dummy_indices()

```
size_t GiNaC::count_dummy_indices (
            const exvector & v)  [inline]
```

Count the number of dummy index pairs in an index vector.

References count_dummy_indices(), and find_free_and_dummy().

Referenced by count_dummy_indices().

7.1.3.174 count_free_indices()

```
size_t GiNaC::count_free_indices (
            const exvector & v)  [inline]
```

Count the number of dummy index pairs in an index vector.

References count_free_indices(), and find_free_and_dummy().

Referenced by count_free_indices().

7.1.3.175 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [11/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            indexed ,
            exprseq ,
            print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_tree > &::do_print_tree )
```

References GiNaC::indexed::do_print(), GiNaC::indexed::do_print_latex(), GiNaC::indexed::do_print_tree(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), and not_symmetric().

7.1.3.176 GINAC_BIND_UNARCHIVER() [21/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            indexed )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.177 indices_consistent()

```
bool GiNaC::indices_consistent (
            const exvector & v1,
            const exvector & v2)  [static]
```

Check whether two sorted index vectors are consistent (i.e.

equal).

References indices_consistent().

Referenced by GiNaC::add::get_free_indices(), indices_consistent(), and GiNaC::indexed::simplify_indexed.

7.1.3.178 number_of_type()

```
template<class T>
size_t GiNaC::number_of_type (
            const exvector & v)
```

References is_exactly_a(), and number_of_type().

Referenced by number_of_type(), and rename_dummy_indices().

7.1.3.179 rename_dummy_indices()

```
template<class T>
ex GiNaC::rename_dummy_indices (
            const ex & e,
            exvector & global_dummy_indices,
            exvector & local_dummy_indices)  [static]
```

Rename dummy indices in an expression.

Parameters

e	Expression to work on
local_dummy_indices	The set of dummy indices that appear in the expression "e"
global_dummy_indices	The set of dummy indices that have appeared before and which we would like to use in "e", too. This gets updated by the function

References GINAC_ASSERT, is_exactly_a(), GiNaC::subs_options::no_pattern, number_of_type(), op(), rename_dummy_indices(), shaker_sort(), and GiNaC::ex::subs().

Referenced by rename_dummy_indices(), GiNaC::indexed::simplify_indexed, and GiNaC::indexed::simplify_indexed_product.

7.1.3.180 find_variant_indices()

```
void GiNaC::find_variant_indices (
            const exvector & v,
            exvector & variant_indices)  [static]
```

Given a set of indices, extract those of class varidx.

References find_variant_indices(), and is_exactly_a().

Referenced by find_variant_indices(), GiNaC::indexed::simplify_indexed, and GiNaC::indexed::simplify_indexed_product.

7.1.3.181 reposition_dummy_indices()

```
bool GiNaC::reposition_dummy_indices (
            ex & e,
            exvector & variant_dummy_indices,
            exvector & moved_indices)
```

Raise/lower dummy indices in a single indexed objects to canonicalize their variance.

Parameters

e	Object to work on
variant_dummy_indices	The set of indices that might need repositioning (will be changed by this function)
moved_indices	The set of indices that have been repositioned (will be changed by this function)

Returns

true if 'e' was changed

7.1.3.182 product_to_exvector()

```
void GiNaC::product_to_exvector (
            const ex & e,
            exvector & v,
            bool & non_commutative)  [static]
```

References _ex2, GINAC_ASSERT, GiNaC::ex::is_equal(), is_exactly_a(), GiNaC::ex::nops(), GiNaC::ex::op(), and product_to_exvector().

Referenced by get_all_dummy_indices(), product_to_exvector(), and GiNaC::indexed::simplify_indexed_product.

7.1.3.183 idx_symmetrization()

```
template<class T>
ex GiNaC::idx_symmetrization (
            const ex & r,
            const exvector & local_dummy_indices)
```

References idx_symmetrization(), is_exactly_a(), GiNaC::ex::nops(), and symmetrize().

Referenced by idx_symmetrization(), GiNaC::indexed::simplify_indexed, and GiNaC::indexed::simplify_indexed_product.

7.1.3.184 simplify_indexed() [3/3]

```
ex GiNaC::simplify_indexed (
            const ex & e,
            exvector & free_indices,
            exvector & dummy_indices,
            const scalar_products & sp)
```

Simplify indexed expression, return list of free indices.

7.1.3.185 simplify_indexed_product()

```
ex GiNaC::simplify_indexed_product (
            const ex & e,
            exvector & free_indices,
            exvector & dummy_indices,
            const scalar_products & sp)
```

Simplify product of indexed expressions (commutative, noncommutative and simple squares), return list of free indices.

7.1.3.186 hasindex()

```
bool GiNaC::hasindex (
            const ex & x,
            const ex & sym)
```

References hasindex(), is_a(), GiNaC::ex::nops(), and GiNaC::ex::op().

Referenced by hasindex(), and GiNaC::indexed::simplify_indexed.

7.1.3.187 get_all_dummy_indices_safely()

```
exvector GiNaC::get_all_dummy_indices_safely (
            const ex & e)
```

More reliable version of the form.

The former assumes that e is an expanded expression.

References ex_to(), find_free_and_dummy(), get_all_dummy_indices_safely(), GiNaC::ex::get_free_indices(), is_a(), GiNaC::ex::nops(), and GiNaC::ex::op().

Referenced by GiNaC::mul::expand(), get_all_dummy_indices_safely(), GiNaC::make_flat_inserter::handle_factor(), rename_dummy_indices_uniquely(), and rename_dummy_indices_uniquely().

7.1.3.188 get_all_dummy_indices()

```
exvector GiNaC::get_all_dummy_indices (
            const ex & e)
```

Returns all dummy indices from the exvector.

Returns all dummy indices from the expression.

References ex_to(), get_all_dummy_indices(), is_a(), and product_to_exvector().

Referenced by expand_dummy_sum(), GiNaC::power::expand_mul(), get_all_dummy_indices(), and get_clifford_comp().

7.1.3.189 rename_dummy_indices_uniquely() [1/4]

```
lst GiNaC::rename_dummy_indices_uniquely (
            const exvector & va,
            const exvector & vb)
```

Similar to above, where va and vb are the same and the return value is a list of two lists for substitution in b.

References dynallocate(), ex_to(), is_a(), is_exactly_a(), and rename_dummy_indices_uniquely().

Referenced by GiNaC::expairseq::construct_from_2_ex(), GiNaC::mul::expand(), GiNaC::ncmul::expand(), GiNaC::power::expand_mul(), GiNaC::make_flat_inserter::handle_factor(), rename_dummy_indices_uniquely(), rename_dummy_indices_uniquely(), rename_dummy_indices_uniquely(), and rename_dummy_indices_uniquely().

7.1.3.190 rename_dummy_indices_uniquely() [2/4]

```
ex GiNaC::rename_dummy_indices_uniquely (
            const exvector & va,
            const exvector & vb,
            const ex & b)
```

Same as above, where va and vb contain the indices of a and b and are sorted.

References ex_to(), GiNaC::subs_options::no_index_renaming, GiNaC::subs_options::no_pattern, GiNaC::ex::nops(), GiNaC::container< C >::op(), rename_dummy_indices_uniquely(), and GiNaC::ex::subs().

7.1.3.191 rename_dummy_indices_uniquely() [3/4]

```
ex GiNaC::rename_dummy_indices_uniquely (
            const ex & a,
            const ex & b)
```

Returns b with all dummy indices, which are common with a, renamed.

References ex_to(), get_all_dummy_indices_safely(), GiNaC::subs_options::no_index_renaming, GiNaC::subs_options::no_pattern, GiNaC::ex::nops(), GiNaC::container< C >::op(), rename_dummy_indices_uniquely(), and GiNaC::ex::subs().

7.1.3.192 rename_dummy_indices_uniquely() [4/4]

```
ex GiNaC::rename_dummy_indices_uniquely (
            exvector & va,
            const ex & b,
            bool modify_va)
```

Returns b with all dummy indices, which are listed in va, renamed if modify_va is set to TRUE all dummy indices of b will be appended to va.

References GiNaC::ex::begin(), GiNaC::ex::end(), ex_to(), get_all_dummy_indices_safely(), GiNaC::subs_options::no_index_renamin GiNaC::subs_options::no_pattern, GiNaC::ex::nops(), GiNaC::container< C >::op(), rename_dummy_indices_uniquely(), and GiNaC::ex::subs().

7.1.3.193 expand_dummy_sum()

```
ex GiNaC::expand_dummy_sum (
            const ex & e,
            bool subs_idx = false)
```

This function returns the given expression with expanded sums for all dummy index summations, where the dimensionality of the dummy index is a nonnegative integer.

Optionally all indices with a variance will be substituted by indices with the corresponding numeric values without variance.

Parameters

e	the given expression
subs_idx	indicates if variance of dummy indices should be neglected

References ex_to(), GiNaC::ex::expand(), expand_dummy_sum(), get_all_dummy_indices(), GiNaC::idx::idx(), is_a(), GiNaC::ex::map(), GiNaC::info_flags::nonnegint, and GiNaC::ex::subs().

Referenced by clifford_to_lst(), expand_dummy_sum(), and remove_dirac_ONE().

7.1.3.194 GINAC_DECLARE_UNARCHIVER() [22/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            indexed )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.195 conjugate_evalf()

```
ex GiNaC::conjugate_evalf (
            const ex & arg)  [static]
```

References conjugate_evalf(), ex_to(), and is_exactly_a().

Referenced by conjugate_evalf(), and REGISTER_FUNCTION().

7.1.3.196 conjugate_eval()

```
ex GiNaC::conjugate_eval (
            const ex & arg)  [static]
```

References GiNaC::ex::conjugate(), and conjugate_eval().

Referenced by conjugate_eval(), and REGISTER_FUNCTION().

7.1.3.197 conjugate_print_latex()

```
void GiNaC::conjugate_print_latex (
            const ex & arg,
            const print_context & c)  [static]
```

References conjugate_print_latex(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by conjugate_print_latex(), and REGISTER_FUNCTION().

7.1.3.198 conjugate_conjugate()

```
ex GiNaC::conjugate_conjugate (
            const ex & arg)  [static]
```

References conjugate_conjugate().

Referenced by conjugate_conjugate(), and REGISTER_FUNCTION().

7.1.3.199 conjugate_expl_derivative()

```
ex GiNaC::conjugate_expl_derivative (
            const ex & arg,
            const symbol & s)  [static]
```

References conjugate(), conjugate_expl_derivative(), GiNaC::ex::diff(), ex_to(), GiNaC::fderivative::fderivative(), GiNaC::symbol::info(), and GiNaC::info_flags::real.

Referenced by conjugate_expl_derivative(), and REGISTER_FUNCTION().

7.1.3.200 conjugate_real_part()

```
ex GiNaC::conjugate_real_part (
            const ex & arg)  [static]
```

References conjugate_real_part(), and GiNaC::ex::real_part().

Referenced by conjugate_real_part(), and REGISTER_FUNCTION().

7.1.3.201 conjugate_imag_part()

```
ex GiNaC::conjugate_imag_part (
            const ex & arg)  [static]
```

References conjugate_imag_part(), and GiNaC::ex::imag_part().

Referenced by conjugate_imag_part(), and REGISTER_FUNCTION().

7.1.3.202 func_arg_info()

```
bool GiNaC::func_arg_info (
            const ex & arg,
            unsigned inf)  [static]
```

References GiNaC::info_flags::cinteger, GiNaC::info_flags::cinteger_polynomial, GiNaC::info_flags::crational, GiNaC::info_flags::crational_polynomial, GiNaC::info_flags::even, func_arg_info(), GiNaC::info_flags::has_indices, GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::info_flags::integer_polynomial, GiNaC::info_flags::negative, GiNaC::info_flags::negint, GiNaC::info_flags::nonnegative, GiNaC::info_flags::nonnegint, GiNaC::info_flags::odd, GiNaC::info_flags::polynomial, GiNaC::info_flags::posint, GiNaC::info_flags::positive, GiNaC::info_flags::prime, GiNaC::info_flags::rational, GiNaC::info_flags::rational_function, GiNaC::info_flags::rational_polynomial, and GiNaC::info_flags::real.

Referenced by conjugate_info(), and func_arg_info().

7.1.3.203 conjugate_info()

```
bool GiNaC::conjugate_info (
            const ex & arg,
            unsigned inf)  [static]
```

References conjugate_info(), and func_arg_info().

Referenced by conjugate_info(), and REGISTER_FUNCTION().

7.1.3.204 REGISTER_FUNCTION() [1/36]

```
GiNaC::REGISTER_FUNCTION (
            conjugate_function ,
            eval_func(conjugate_eval).  evalf_func(conjugate_evalf).  expl_derivative_↩
func(conjugate_expl_derivative).  info_func(conjugate_info).  print_func< print_latex >(conjugate_print_latex
conjugate_func(conjugate_conjugate).  real_part_func(conjugate_real_part).  imag_part_func(conjugate_imag_part
set_name("conjugate","conjugate") )
```

References conjugate_conjugate(), conjugate_eval(), conjugate_evalf(), conjugate_expl_derivative(), conjugate_imag_part(), conjugate_info(), conjugate_print_latex(), conjugate_real_part(), and REGISTER_FUNCTION().

Referenced by REGISTER_FUNCTION(), and REGISTER_FUNCTION().

7.1.3.205 real_part_evalf()

```
ex GiNaC::real_part_evalf (
            const ex & arg)  [static]
```

References ex_to(), is_exactly_a(), and real_part_evalf().

Referenced by real_part_evalf(), and REGISTER_FUNCTION().

7.1.3.206 real_part_eval()

```
ex GiNaC::real_part_eval (
            const ex & arg)  [static]
```

References GiNaC::ex::real_part(), and real_part_eval().

Referenced by real_part_eval(), and REGISTER_FUNCTION().

7.1.3.207 real_part_print_latex()

```
void GiNaC::real_part_print_latex (
            const ex & arg,
            const print_context & c)  [static]
```

References GiNaC::ex::print(), real_part_print_latex(), and GiNaC::print_context::s.

Referenced by real_part_print_latex(), and REGISTER_FUNCTION().

7.1.3.208 real_part_conjugate()

```
ex GiNaC::real_part_conjugate (
            const ex & arg)  [static]
```

References real_part_conjugate().

Referenced by real_part_conjugate(), and REGISTER_FUNCTION().

7.1.3.209 real_part_real_part()

```
ex GiNaC::real_part_real_part (
            const ex & arg)  [static]
```

References real_part_real_part().

Referenced by real_part_real_part(), and REGISTER_FUNCTION().

7.1.3.210 real_part_imag_part()

```
ex GiNaC::real_part_imag_part (
            const ex & arg)  [static]
```

References real_part_imag_part().

Referenced by real_part_imag_part(), and REGISTER_FUNCTION().

7.1.3.211 real_part_expl_derivative()

```
ex GiNaC::real_part_expl_derivative (
            const ex & arg,
            const symbol & s)  [static]
```

References GiNaC::ex::diff(), ex_to(), GiNaC::fderivative::fderivative(), GiNaC::symbol::info(), GiNaC::info_flags::real, real_part(), and real_part_expl_derivative().

Referenced by real_part_expl_derivative(), and REGISTER_FUNCTION().

7.1.3.212 REGISTER_FUNCTION() [2/36]

```
GiNaC::REGISTER_FUNCTION (
            real_part_function ,
            eval_func(real_part_eval).  evalf_func(real_part_evalf).  expl_derivative_↩
func(real_part_expl_derivative).  print_func< print_latex >(real_part_print_latex).  conjugate↩
_func(real_part_conjugate).  real_part_func(real_part_real_part).  imag_part_func(real_part_imag_part).
set_name("real_part","real_part") )
```

References real_part_conjugate(), real_part_eval(), real_part_evalf(), real_part_expl_derivative(), real_part_imag_part(), real_part_print_latex(), real_part_real_part(), and REGISTER_FUNCTION().

7.1.3.213 imag_part_evalf()

```
ex GiNaC::imag_part_evalf (
            const ex & arg)  [static]
```

References ex_to(), imag_part_evalf(), and is_exactly_a().

Referenced by imag_part_evalf(), and REGISTER_FUNCTION().

7.1.3.214 imag_part_eval()

```
ex GiNaC::imag_part_eval (
            const ex & arg)  [static]
```

References GiNaC::ex::imag_part(), and imag_part_eval().

Referenced by imag_part_eval(), and REGISTER_FUNCTION().

7.1.3.215 imag_part_print_latex()

```
void GiNaC::imag_part_print_latex (
            const ex & arg,
            const print_context & c)  [static]
```

References imag_part_print_latex(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by imag_part_print_latex(), and REGISTER_FUNCTION().

7.1.3.216 imag_part_conjugate()

```
ex GiNaC::imag_part_conjugate (
            const ex & arg)  [static]
```

References imag_part_conjugate().

Referenced by imag_part_conjugate(), and REGISTER_FUNCTION().

7.1.3.217 imag_part_real_part()

```
ex GiNaC::imag_part_real_part (
            const ex & arg)  [static]
```

References imag_part_real_part().

Referenced by imag_part_real_part(), and REGISTER_FUNCTION().

7.1.3.218 imag_part_imag_part()

```
ex GiNaC::imag_part_imag_part (
            const ex & arg)  [static]
```

References imag_part_imag_part().

Referenced by imag_part_imag_part(), and REGISTER_FUNCTION().

7.1.3.219 imag_part_expl_derivative()

```
ex GiNaC::imag_part_expl_derivative (
            const ex & arg,
            const symbol & s)  [static]
```

References GiNaC::ex::diff(), ex_to(), GiNaC::fderivative::fderivative(), imag_part(), imag_part_expl_derivative(), GiNaC::symbol::info(), and GiNaC::info_flags::real.

Referenced by imag_part_expl_derivative(), and REGISTER_FUNCTION().

7.1.3.220 REGISTER_FUNCTION() [3/36]

```
GiNaC::REGISTER_FUNCTION (
            imag_part_function ,
            eval_func(imag_part_eval).  evalf_func(imag_part_evalf).  expl_derivative_↩
func(imag_part_expl_derivative).  print_func< print_latex >(imag_part_print_latex).  conjugate↩
_func(imag_part_conjugate).  real_part_func(imag_part_real_part).  imag_part_func(imag_part_imag_part).
set_name("imag_part","imag_part") )
```

References imag_part_conjugate(), imag_part_eval(), imag_part_evalf(), imag_part_expl_derivative(), imag_part_imag_part(), imag_part_print_latex(), imag_part_real_part(), and REGISTER_FUNCTION().

7.1.3.221 abs_evalf()

```
ex GiNaC::abs_evalf (
            const ex & arg)  [static]
```

References abs(), abs_evalf(), ex_to(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by abs_evalf(), and REGISTER_FUNCTION().

7.1.3.222 abs_eval()

```
ex GiNaC::abs_eval (
            const ex & arg)  [static]
```

References abs(), abs_eval(), ex_to(), exp(), GiNaC::basic::hold(), GiNaC::ex::info(), is_ex_the_function, is_exactly_a(), GiNaC::info_flags::negative, GiNaC::info_flags::nonnegative, GiNaC::ex::op(), GiNaC::info_flags::positive, pow(), GiNaC::info_flags::real, GiNaC::ex::real_part(), and step().

Referenced by abs_eval(), and REGISTER_FUNCTION().

7.1.3.223 abs_expand()

```
ex GiNaC::abs_expand (
            const ex & arg,
            unsigned options)  [static]
```

References abs(), abs_expand(), GiNaC::ex::begin(), dynallocate(), GiNaC::ex::end(), GiNaC::ex::expand(), GiNaC::expand_options::expand_function_args, GiNaC::expand_options::expand_transcendental, GiNaC::status_flags::expanded, GiNaC::basic::hold(), is_exactly_a(), and GiNaC::ex::nops().

Referenced by abs_expand(), and REGISTER_FUNCTION().

7.1.3.224 abs_expl_derivative()

```
ex GiNaC::abs_expl_derivative (
            const ex & arg,
            const symbol & s)  [static]
```

References abs(), abs_expl_derivative(), GiNaC::ex::conjugate(), and GiNaC::ex::diff().

Referenced by abs_expl_derivative(), and REGISTER_FUNCTION().

7.1.3.225 abs_print_latex()

```
void GiNaC::abs_print_latex (
            const ex & arg,
            const print_context & c)  [static]
```

References abs_print_latex(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by abs_print_latex(), and REGISTER_FUNCTION().

7.1.3.226 abs_print_csrc_float()

```
void GiNaC::abs_print_csrc_float (
            const ex & arg,
            const print_context & c)  [static]
```

References abs_print_csrc_float(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by abs_print_csrc_float(), and REGISTER_FUNCTION().

7.1.3.227 abs_conjugate()

```
ex GiNaC::abs_conjugate (
            const ex & arg)  [static]
```

References abs(), abs_conjugate(), and GiNaC::basic::hold().

Referenced by abs_conjugate(), and REGISTER_FUNCTION().

7.1.3.228 abs_real_part()

```
ex GiNaC::abs_real_part (
            const ex & arg)  [static]
```

References abs(), abs_real_part(), and GiNaC::basic::hold().

Referenced by abs_real_part(), and REGISTER_FUNCTION().

7.1.3.229 abs_imag_part()

```
ex GiNaC::abs_imag_part (
            const ex & arg)  [static]
```

References abs_imag_part().

Referenced by abs_imag_part(), and REGISTER_FUNCTION().

7.1.3.230 abs_power()

```
ex GiNaC::abs_power (
            const ex & arg,
            const ex & exp)  [static]
```

References abs(), abs_power(), GiNaC::ex::conjugate(), GiNaC::info_flags::even, ex_to(), exp(), GiNaC::ex::info(), GiNaC::numeric::info(), is_a(), GiNaC::ex::is_equal(), is_even(), pow(), GiNaC::power::power(), and GiNaC::info_flags::real.

Referenced by abs_power(), and REGISTER_FUNCTION().

7.1.3.231 abs_info()

```
bool GiNaC::abs_info (
            const ex & arg,
            unsigned inf)
```

References abs_info(), GiNaC::info_flags::even, GiNaC::info_flags::has_indices, GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::info_flags::negative, GiNaC::info_flags::nonnegative, GiNaC::info_flags::nonnegint, GiNaC::info_flags::odd, GiNaC::info_flags::positive, GiNaC::info_flags::prime, and GiNaC::info_flags::real.

Referenced by abs_info(), and REGISTER_FUNCTION().

7.1.3.232 REGISTER_FUNCTION() [4/36]

```
GiNaC::REGISTER_FUNCTION (
            abs ,
            eval_func(abs_eval).  evalf_func(abs_evalf).  expand_func(abs_expand).  expl_↩
derivative_func(abs_expl_derivative).  info_func(abs_info).  print_func< print_latex >(abs_print_latex).
print_func< print_csrc_float >(abs_print_csrc_float).  print_func< print_csrc_double >(abs_print_csrc_float)
conjugate_func(abs_conjugate).  real_part_func(abs_real_part).  imag_part_func(abs_imag_part).
power_func(abs_power) )
```

References abs(), abs_conjugate(), abs_eval(), abs_evalf(), abs_expand(), abs_expl_derivative(), abs_imag_part(), abs_info(), abs_power(), abs_print_csrc_float(), abs_print_latex(), abs_real_part(), and REGISTER_FUNCTION().

7.1.3.233 step_evalf()

```
ex GiNaC::step_evalf (
            const ex & arg)  [static]
```

References ex_to(), GiNaC::basic::hold(), is_exactly_a(), step(), and step_evalf().

Referenced by REGISTER_FUNCTION(), and step_evalf().

7.1.3.234 step_eval()

```
ex GiNaC::step_eval (
            const ex & arg)  [static]
```

References ex_to(), GiNaC::basic::hold(), I, GiNaC::numeric::imag(), is_exactly_a(), GiNaC::numeric::is_real(), GiNaC::numeric::is_zero(), GiNaC::ex::nops(), GiNaC::ex::op(), GiNaC::numeric::real(), step(), and step_eval().

Referenced by REGISTER_FUNCTION(), and step_eval().

7.1.3.235 step_series()

```
ex GiNaC::step_series (
            const ex & arg,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex0, ex_to(), GiNaC::expair::expair(), GiNaC::ex::info(), GiNaC::subs_options::no_pattern, GiNaC::info_flags::numeric, GiNaC::pseries::pseries(), step(), step_series(), GiNaC::ex::subs(), and GiNaC::series_options::suppress

Referenced by REGISTER_FUNCTION(), and step_series().

7.1.3.236 step_conjugate()

```
ex GiNaC::step_conjugate (
            const ex & arg)  [static]
```

References GiNaC::basic::hold(), step(), and step_conjugate().

Referenced by REGISTER_FUNCTION(), and step_conjugate().

7.1.3.237 step_real_part()

```
ex GiNaC::step_real_part (
            const ex & arg)  [static]
```

References GiNaC::basic::hold(), step(), and step_real_part().

Referenced by REGISTER_FUNCTION(), and step_real_part().

7.1.3.238 step_imag_part()

```
ex GiNaC::step_imag_part (
            const ex & arg)  [static]
```

References step_imag_part().

Referenced by REGISTER_FUNCTION(), and step_imag_part().

7.1.3.239 REGISTER_FUNCTION() [5/36]

```
GiNaC::REGISTER_FUNCTION (
            step ,
            eval_func(step_eval).  evalf_func(step_evalf).  series_func(step_series).  conjugate↩
_func(step_conjugate).  real_part_func(step_real_part).  imag_part_func(step_imag_part) )
```

References REGISTER_FUNCTION(), step(), step_conjugate(), step_eval(), step_evalf(), step_imag_part(), step_real_part(), and step_series().

7.1.3.240 csgn_evalf()

```
ex GiNaC::csgn_evalf (
            const ex & arg)  [static]
```

References csgn(), csgn_evalf(), ex_to(), and is_exactly_a().

Referenced by csgn_evalf(), and REGISTER_FUNCTION().

7.1.3.241 csgn_eval()

```
ex GiNaC::csgn_eval (
            const ex & arg)  [static]
```

References csgn(), csgn_eval(), ex_to(), I, GiNaC::numeric::imag(), is_exactly_a(), GiNaC::numeric::is_real(), GiNaC::numeric::is_zero(), GiNaC::ex::nops(), GiNaC::ex::op(), and GiNaC::numeric::real().

Referenced by csgn_eval(), and REGISTER_FUNCTION().

7.1.3.242 csgn_series()

```
ex GiNaC::csgn_series (
            const ex & arg,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex0, csgn(), csgn_series(), ex_to(), GiNaC::expair::expair(), GiNaC::ex::info(), GiNaC::subs_options::no_pattern, GiNaC::info_flags::numeric, GiNaC::pseries::pseries(), GiNaC::ex::subs(), and GiNaC::series_options::suppress_branchcut.

Referenced by csgn_series(), and REGISTER_FUNCTION().

7.1.3.243 csgn_conjugate()

```
ex GiNaC::csgn_conjugate (
            const ex & arg)  [static]
```

References csgn(), and csgn_conjugate().

Referenced by csgn_conjugate(), and REGISTER_FUNCTION().

7.1.3.244 csgn_real_part()

```
ex GiNaC::csgn_real_part (
            const ex & arg)  [static]
```

References csgn(), and csgn_real_part().

Referenced by csgn_real_part(), and REGISTER_FUNCTION().

7.1.3.245 csgn_imag_part()

```
ex GiNaC::csgn_imag_part (
            const ex & arg)  [static]
```

References csgn_imag_part().

Referenced by csgn_imag_part(), and REGISTER_FUNCTION().

7.1.3.246 csgn_power()

```
ex GiNaC::csgn_power (
            const ex & arg,
            const ex & exp)  [static]
```

References _ex2, csgn(), csgn_power(), ex_to(), exp(), GiNaC::numeric::info(), is_a(), is_odd(), GiNaC::info_flags::positive, and GiNaC::power::power().

Referenced by csgn_power(), and REGISTER_FUNCTION().

7.1.3.247 REGISTER_FUNCTION() [6/36]

```
GiNaC::REGISTER_FUNCTION (
            csgn ,
            eval_func(csgn_eval).  evalf_func(csgn_evalf).  series_func(csgn_series).  conjugate↩
_func(csgn_conjugate).  real_part_func(csgn_real_part).  imag_part_func(csgn_imag_part).  power↩
_func(csgn_power) )
```

References csgn(), csgn_conjugate(), csgn_eval(), csgn_evalf(), csgn_imag_part(), csgn_power(), csgn_real_part(), csgn_series(), and REGISTER_FUNCTION().

7.1.3.248 eta_evalf()

```
ex GiNaC::eta_evalf (
            const ex & x,
            const ex & y)  [static]
```

References _ex0, csgn(), eta_evalf(), evalf(), ex_to(), I, imag(), GiNaC::ex::info(), GiNaC::numeric::is_negative(), GiNaC::numeric::is_real(), GiNaC::info_flags::numeric, Pi, and GiNaC::info_flags::positive.

Referenced by eta_evalf(), and REGISTER_FUNCTION().

7.1.3.249 eta_eval()

```
ex GiNaC::eta_eval (
            const ex & x,
            const ex & y)  [static]
```

References _ex0, csgn(), eta_eval(), ex_to(), I, imag(), GiNaC::ex::info(), GiNaC::numeric::is_negative(), GiNaC::numeric::is_real(), GiNaC::info_flags::numeric, Pi, and GiNaC::info_flags::positive.

Referenced by eta_eval(), and REGISTER_FUNCTION().

7.1.3.250 eta_series()

```
ex GiNaC::eta_series (
            const ex & x,
            const ex & y,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex0, eta_series(), GiNaC::expair::expair(), GiNaC::ex::info(), GiNaC::info_flags::negative, GiNaC::subs_options::no_patte
GiNaC::info_flags::numeric, GiNaC::pseries::pseries(), and GiNaC::ex::subs().

Referenced by eta_series(), and REGISTER_FUNCTION().

7.1.3.251 eta_conjugate()

```
ex GiNaC::eta_conjugate (
            const ex & x,
            const ex & y)  [static]
```

References eta_conjugate().

Referenced by eta_conjugate(), and REGISTER_FUNCTION().

7.1.3.252 eta_real_part()

```
ex GiNaC::eta_real_part (
            const ex & x,
            const ex & y)  [static]
```

References eta_real_part().

Referenced by eta_real_part(), and REGISTER_FUNCTION().

7.1.3.253 eta_imag_part()

```
ex GiNaC::eta_imag_part (
            const ex & x,
            const ex & y)  [static]
```

References eta_imag_part(), GiNaC::basic::hold(), and I.

Referenced by eta_imag_part(), and REGISTER_FUNCTION().

7.1.3.254 REGISTER_FUNCTION() [7/36]

```
GiNaC::REGISTER_FUNCTION (
            eta ,
            eval_func(eta_eval).  evalf_func(eta_evalf).  series_func(eta_series).  latex↩
_name("\\eta").  set_symmetry(sy_symm(0, 1)).  conjugate_func(eta_conjugate).  real_part_↩
func(eta_real_part).  imag_part_func(eta_imag_part) )
```

References eta_conjugate(), eta_eval(), eta_evalf(), eta_imag_part(), eta_real_part(), eta_series(), REGISTER_FUNCTION(), and sy_symm().

7.1.3.255 Li2_evalf()

```
ex GiNaC::Li2_evalf (
            const ex & x)  [static]
```

References ex_to(), GiNaC::basic::hold(), is_exactly_a(), Li2(), and Li2_evalf().

Referenced by Li2_evalf(), and REGISTER_FUNCTION().

7.1.3.256 Li2_eval()

```
ex GiNaC::Li2_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex12, _ex1_2, _ex2, _ex6, _ex_1, _ex_1_2, _ex_48, Catalan, GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), I, GiNaC::ex::info(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), Li2(), Li2_eval(), log(), GiNaC::info_flags::numeric, Pi, and GiNaC::power::power().

Referenced by Li2_eval(), and REGISTER_FUNCTION().

7.1.3.257 Li2_deriv()

```
ex GiNaC::Li2_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex1, GINAC_ASSERT, Li2_deriv(), and log().

Referenced by Li2_deriv(), and REGISTER_FUNCTION().

7.1.3.258 Li2_series() [1/2]

```
ex GiNaC::Li2_series (
             const ex & x,
             const relational & rel,
             int order,
             unsigned options)  [static]
```

References _ex0, _ex1, _ex2, _num2_p, ex_to(), GiNaC::expair::expair(), I, GiNaC::ex::info(), GiNaC::ex::is_equal(), is_real(), GiNaC::ex::is_zero(), GiNaC::relational::lhs(), Li2(), Li2_series(), log(), GiNaC::subs_options::no_pattern, GiNaC::ex::nops(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), GiNaC::ex::op(), op(), Pi, pow(), GiNaC::pseries::pseries(), GiNaC::relational::rhs(), GiNaC::ex::series(), series(), GiNaC::ex::subs(), subs(), GiNaC::series_options::suppress_branchcut, and zeta().

Referenced by Li2_projection(), Li2_series(), Li2_series(), and REGISTER_FUNCTION().

7.1.3.259 Li2_conjugate()

```
ex GiNaC::Li2_conjugate (
             const ex & x)  [static]
```

References _num1_p, GiNaC::ex::conjugate(), GiNaC::basic::hold(), GiNaC::ex::imag_part(), GiNaC::ex::info(), is_exactly_a(), GiNaC::ex::is_zero(), Li2(), Li2_conjugate(), and GiNaC::info_flags::negative.

Referenced by Li2_conjugate(), and REGISTER_FUNCTION().

7.1.3.260 REGISTER_FUNCTION() [8/36]

```
GiNaC::REGISTER_FUNCTION (
             Li2 ,
             eval_func(Li2_eval).  evalf_func(Li2_evalf).  derivative_func(Li2_deriv).  series↩
_func(Li2_series).  conjugate_func(Li2_conjugate).  latex_name("\\mathrm{Li}_2") )
```

References Li2(), Li2_conjugate(), Li2_deriv(), Li2_eval(), Li2_evalf(), Li2_series(), and REGISTER_FUNCTION().

7.1.3.261 Li3_eval()

```
ex GiNaC::Li3_eval (
             const ex & x)  [static]
```

References GiNaC::ex::is_zero(), and Li3_eval().

Referenced by Li3_eval(), and REGISTER_FUNCTION().

7.1.3.262 REGISTER_FUNCTION() [9/36]

```
GiNaC::REGISTER_FUNCTION (
             Li3 ,
             eval_func(Li3_eval).  latex_name("\\mathrm{Li}_3") )
```

References Li3_eval(), and REGISTER_FUNCTION().

7.1.3.263 zetaderiv_eval()

```
ex GiNaC::zetaderiv_eval (
             const ex & n,
             const ex & x)  [static]
```

References GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::ex::is_zero(), GiNaC::info_flags::numeric, zeta(), and zetaderiv_eval().

Referenced by REGISTER_FUNCTION(), and zetaderiv_eval().

7.1.3.264 zetaderiv_deriv()

```
ex GiNaC::zetaderiv_deriv (
             const ex & n,
             const ex & x,
             unsigned deriv_param)  [static]
```

References GINAC_ASSERT, and zetaderiv_deriv().

Referenced by REGISTER_FUNCTION(), and zetaderiv_deriv().

7.1.3.265 REGISTER_FUNCTION() [10/36]

```
GiNaC::REGISTER_FUNCTION (
             zetaderiv ,
             eval_func(zetaderiv_eval).  derivative_func(zetaderiv_deriv).  latex_name("\\zeta^\\prime")
)
```

References REGISTER_FUNCTION(), zetaderiv_deriv(), and zetaderiv_eval().

7.1.3.266 factorial_evalf()

```
ex GiNaC::factorial_evalf (
             const ex & x)  [static]
```

References factorial(), factorial_evalf(), and GiNaC::basic::hold().

Referenced by factorial_evalf(), and REGISTER_FUNCTION().

7.1.3.267 factorial_eval()

```
ex GiNaC::factorial_eval (
             const ex & x)  [static]
```

References ex_to(), factorial(), factorial_eval(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by factorial_eval(), and REGISTER_FUNCTION().

7.1.3.268 factorial_print_dflt_latex()

```
void GiNaC::factorial_print_dflt_latex (
            const ex & x,
            const print_context & c)  [static]
```

References factorial_print_dflt_latex(), is_exactly_a(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by factorial_print_dflt_latex(), and REGISTER_FUNCTION().

7.1.3.269 factorial_conjugate()

```
ex GiNaC::factorial_conjugate (
            const ex & x)  [static]
```

References factorial(), factorial_conjugate(), and GiNaC::basic::hold().

Referenced by factorial_conjugate(), and REGISTER_FUNCTION().

7.1.3.270 factorial_real_part()

```
ex GiNaC::factorial_real_part (
            const ex & x)  [static]
```

References factorial(), factorial_real_part(), and GiNaC::basic::hold().

Referenced by factorial_real_part(), and REGISTER_FUNCTION().

7.1.3.271 factorial_imag_part()

```
ex GiNaC::factorial_imag_part (
            const ex & x)  [static]
```

References factorial_imag_part().

Referenced by factorial_imag_part(), and REGISTER_FUNCTION().

7.1.3.272 REGISTER_FUNCTION() [11/36]

```
GiNaC::REGISTER_FUNCTION (
            factorial ,
            eval_func(factorial_eval).  evalf_func(factorial_evalf).  print_func< print_dflt
>(factorial_print_dflt_latex).  print_func< print_latex >(factorial_print_dflt_latex).  conjugate↩
_func(factorial_conjugate).  real_part_func(factorial_real_part).  imag_part_func(factorial_imag_part)
)
```

References factorial(), factorial_conjugate(), factorial_eval(), factorial_evalf(), factorial_imag_part(), factorial_print_dflt_latex(), factorial_real_part(), and REGISTER_FUNCTION().

7.1.3.273 binomial_evalf()

```
ex GiNaC::binomial_evalf (
             const ex & x,
             const ex & y)  [static]
```

References binomial(), binomial_evalf(), and GiNaC::basic::hold().

Referenced by binomial_evalf(), and REGISTER_FUNCTION().

7.1.3.274 binomial_sym()

```
ex GiNaC::binomial_sym (
             const ex & x,
             const numeric & y)  [static]
```

References _ex0, _ex1, binomial(), binomial_sym(), GiNaC::ex::expand(), GiNaC::basic::hold(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_nonneg_integer(), and GiNaC::numeric::to_int().

Referenced by binomial_eval(), and binomial_sym().

7.1.3.275 binomial_eval()

```
ex GiNaC::binomial_eval (
             const ex & x,
             const ex & y)  [static]
```

References binomial(), binomial_eval(), binomial_sym(), ex_to(), GiNaC::basic::hold(), is_exactly_a(), and is_integer().

Referenced by binomial_eval(), and REGISTER_FUNCTION().

7.1.3.276 binomial_conjugate()

```
ex GiNaC::binomial_conjugate (
             const ex & x,
             const ex & y)  [static]
```

References binomial(), binomial_conjugate(), and GiNaC::basic::hold().

Referenced by binomial_conjugate(), and REGISTER_FUNCTION().

7.1.3.277 binomial_real_part()

```
ex GiNaC::binomial_real_part (
             const ex & x,
             const ex & y)  [static]
```

References binomial(), binomial_real_part(), and GiNaC::basic::hold().

Referenced by binomial_real_part(), and REGISTER_FUNCTION().

7.1.3.278 binomial_imag_part()

```
ex GiNaC::binomial_imag_part (
            const ex & x,
            const ex & y)  [static]
```

References binomial_imag_part().

Referenced by binomial_imag_part(), and REGISTER_FUNCTION().

7.1.3.279 REGISTER_FUNCTION() [12/36]

```
GiNaC::REGISTER_FUNCTION (
            binomial ,
            eval_func(binomial_eval).  evalf_func(binomial_evalf).  conjugate_func(binomial_conjugate).
real_part_func(binomial_real_part).  imag_part_func(binomial_imag_part) )
```

References binomial(), binomial_conjugate(), binomial_eval(), binomial_evalf(), binomial_imag_part(), binomial_real_part(), and REGISTER_FUNCTION().

7.1.3.280 Order_eval()

```
ex GiNaC::Order_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, ex_to(), is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::expairseq::nops(), GiNaC::expairseq::op(), and Order_eval().

Referenced by Order_eval(), and REGISTER_FUNCTION().

7.1.3.281 Order_series()

```
ex GiNaC::Order_series (
            const ex & x,
            const relational & r,
            int order,
            unsigned options)  [static]
```

References _ex1, ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, is_a(), GiNaC::ex::ldegree(), GiNaC::relational::lhs(), GiNaC::numeric::numeric(), Order_series(), and GiNaC::pseries::pseries().

Referenced by Order_series(), and REGISTER_FUNCTION().

7.1.3.282 Order_conjugate()

```
ex GiNaC::Order_conjugate (
            const ex & x)  [static]
```

References Order_conjugate().

Referenced by Order_conjugate(), and REGISTER_FUNCTION().

7.1.3.283 Order_real_part()

```
ex GiNaC::Order_real_part (
            const ex & x)  [static]
```

References Order_real_part().

Referenced by Order_real_part(), and REGISTER_FUNCTION().

7.1.3.284 Order_imag_part()

```
ex GiNaC::Order_imag_part (
            const ex & x)  [static]
```

References GiNaC::ex::info(), Order_imag_part(), and GiNaC::info_flags::real.

Referenced by Order_imag_part(), and REGISTER_FUNCTION().

7.1.3.285 Order_power()

```
ex GiNaC::Order_power (
            const ex & x,
            const ex & e)  [static]
```

References GiNaC::ex::info(), is_exactly_a(), Order_power(), GiNaC::info_flags::posint, pow(), and GiNaC::power::power().

Referenced by Order_power(), and REGISTER_FUNCTION().

7.1.3.286 Order_expl_derivative()

```
ex GiNaC::Order_expl_derivative (
            const ex & arg,
            const symbol & s)  [static]
```

References GiNaC::ex::diff(), and Order_expl_derivative().

Referenced by Order_expl_derivative(), and REGISTER_FUNCTION().

7.1.3.287 REGISTER_FUNCTION() [13/36]

```
GiNaC::REGISTER_FUNCTION (
            Order ,
            eval_func(Order_eval).  series_func(Order_series).  latex_name("\\mathcal{O}").
expl_derivative_func(Order_expl_derivative).  conjugate_func(Order_conjugate).  real_part_↩
func(Order_real_part).  imag_part_func(Order_imag_part).  power_func(Order_power) )
```

References Order_conjugate(), Order_eval(), Order_expl_derivative(), Order_imag_part(), Order_power(), Order_real_part(), Order_series(), and REGISTER_FUNCTION().

7.1.3.288 lsolve()

```
ex GiNaC::lsolve (
              const ex & eqns,
              const ex & symbols,
              unsigned options = solve_algo::automatic)
```

Factorial function.

Binomial function. Order term function (for truncated power series).

References GiNaC::container< C >::append(), GiNaC::ex::coeff(), GiNaC::matrix::cols(), ex_to(), GiNaC::ex::expand(), GiNaC::info_flags::exprseq, GINAC_ASSERT, GiNaC::symbolset::has(), GiNaC::ex::info(), is_exactly_a(), GiNaC::info_flags::list, lsolve(), GiNaC::ex::nops(), GiNaC::ex::op(), GiNaC::info_flags::relation_equal, rhs(), GiNaC::matrix::rows(), GiNaC::matrix::solve(), and GiNaC::info_flags::symbol.

Referenced by lsolve().

7.1.3.289 fsolve()

```
const numeric GiNaC::fsolve (
              const ex & f,
              const symbol & x,
              const numeric & x1,
              const numeric & x2)
```

Find a real root of real-valued function f(x) numerically within a given interval.

The function must change sign across interval. Uses Newton- Raphson method combined with bisection in order to guarantee convergence.

Parameters

f	Function f(x)
x	Symbol f(x)
x1	lower interval limit
x2	upper interval limit

Exceptions

runtime_error	(if interval is invalid).

References GiNaC::ex::diff(), GiNaC::ex::evalf(), ex_to(), fsolve(), is_a(), is_real(), GiNaC::numeric::is_real(), GiNaC::numeric::is_zero(), GiNaC::ex::lhs(), normal(), GiNaC::ex::rhs(), and GiNaC::ex::subs().

Referenced by fsolve().

7.1.3.290 zeta() [1/3]

```
template<typename T1>
function GiNaC::zeta (
            const T1 & p1)  [inline]
```

References GiNaC::ex::ex(), GiNaC::function::function(), GiNaC::zeta1_SERIAL::serial, and zeta().

Referenced by GINAC_DECLARE_UNARCHIVER(), Li2_series(), Li_eval(), psi2_eval(), GiNaC::Gt::qExpand(), S_eval(), zeta(), zeta(), zeta(), zeta1_eval(), zeta1_evalf(), zeta2_eval(), zeta2_evalf(), and zetaderiv_eval().

7.1.3.291 zeta() [2/3]

```
template<typename T1, typename T2>
function GiNaC::zeta (
            const T1 & p1,
            const T2 & p2)  [inline]
```

References GiNaC::ex::ex(), GiNaC::function::function(), GiNaC::zeta2_SERIAL::serial, and zeta().

7.1.3.292 is_the_function< zeta_SERIAL >()

```
template<>
bool GiNaC::is_the_function< zeta_SERIAL > (
            const ex & x)  [inline]
```

References is_the_function().

7.1.3.293 G() [1/2]

```
template<typename T1, typename T2>
function GiNaC::G (
            const T1 & x,
            const T2 & y)  [inline]
```

References GiNaC::ex::ex(), GiNaC::function::function(), G(), and GiNaC::G2_SERIAL::serial.

Referenced by clifford_moebius_map(), clifford_moebius_map(), G(), G(), G2_eval(), G2_evalf(), G3_eval(), G3_evalf(), and GiNaC::Gt_detail::pathintegral_term::G_path().

7.1.3.294 G() [2/2]

```
template<typename T1, typename T2, typename T3>
function GiNaC::G (
            const T1 & x,
            const T2 & s,
            const T3 & y)  [inline]
```

References GiNaC::ex::ex(), GiNaC::function::function(), G(), and GiNaC::G3_SERIAL::serial.

7.1.3.295 is_the_function< G_SERIAL >()

```
template<>
bool GiNaC::is_the_function< G_SERIAL > (
             const ex & x)  [inline]
```

References is_the_function().

7.1.3.296 psi() [1/4]

```
template<typename T1>
function GiNaC::psi (
             const T1 & p1)  [inline]
```

References GiNaC::ex::ex(), GiNaC::function::function(), psi(), and GiNaC::psi1_SERIAL::serial.

Referenced by beta_deriv(), GINAC_DECLARE_UNARCHIVER(), lgamma_deriv(), psi(), psi(), psi(), psi(), psi1_deriv(), psi1_eval(), psi1_evalf(), psi1_series(), psi2_deriv(), psi2_eval(), psi2_evalf(), psi2_series(), sr_gcd(), and tgamma_deriv().

7.1.3.297 psi() [2/4]

```
template<typename T1, typename T2>
function GiNaC::psi (
             const T1 & p1,
             const T2 & p2)  [inline]
```

References GiNaC::ex::ex(), GiNaC::function::function(), psi(), and GiNaC::psi2_SERIAL::serial.

7.1.3.298 is_the_function< psi_SERIAL >()

```
template<>
bool GiNaC::is_the_function< psi_SERIAL > (
             const ex & x)  [inline]
```

References is_the_function().

7.1.3.299 iterated_integral() [1/2]

```
template<typename T1, typename T2>
function GiNaC::iterated_integral (
             const T1 & kernel_lst,
             const T2 & lambda)  [inline]
```

References GiNaC::ex::ex(), GiNaC::function::function(), iterated_integral(), and GiNaC::iterated_integral2_SERIAL::serial.

Referenced by iterated_integral(), iterated_integral(), iterated_integral2_eval(), iterated_integral3_eval(), and iterated_integral_evalf_impl().

7.1.3.300 iterated_integral() [2/2]

```
template<typename T1, typename T2, typename T3>
function GiNaC::iterated_integral (
             const T1 & kernel_lst,
             const T2 & lambda,
             const T3 & N_trunc)  [inline]
```

References GiNaC::ex::ex(), GiNaC::function::function(), iterated_integral(), and GiNaC::iterated_integral3_SERIAL::serial.

7.1.3.301 is_the_function< iterated_integral_SERIAL >()

```
template<>
bool GiNaC::is_the_function< iterated_integral_SERIAL > (
             const ex & x)  [inline]
```

References is_the_function().

7.1.3.302 is_order_function()

```
bool GiNaC::is_order_function (
             const ex & e)  [inline]
```

Check whether a function is the Order (O(n)) function.

References is_ex_the_function, and is_order_function().

Referenced by GiNaC::pseries::add_series(), GiNaC::pseries::convert_to_poly(), GiNaC::pseries::derivative(), is_order_function(), GiNaC::pseries::is_terminating(), GiNaC::pseries::mul_const(), GiNaC::pseries::mul_series(), GiNaC::pseries::op(), GiNaC::pseries::power_const(), GiNaC::pseries::print_series(), and GiNaC::integral::series().

7.1.3.303 convert_H_to_Li()

```
ex GiNaC::convert_H_to_Li (
             const ex & parameterlst,
             const ex & arg)
```

Converts a given list containing parameters for H in Remiddi/Vermaseren notation into the corresponding GiNaC functions.

References convert_H_to_Li(), and is_a().

Referenced by convert_H_to_Li().

7.1.3.304 EllipticK_evalf()

```
ex GiNaC::EllipticK_evalf (
             const ex & k)  [static]
```

References EllipticK_evalf(), GiNaC::ex::evalf(), ex_to(), GiNaC::ex::info(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), Pi, and sqrt().

Referenced by EllipticK_evalf(), and REGISTER_FUNCTION().

7.1.3.305 EllipticK_eval()

```
ex GiNaC::EllipticK_eval (
            const ex & k)  [static]
```

References _ex0, GiNaC::info_flags::crational, EllipticK_eval(), GiNaC::ex::info(), GiNaC::info_flags::numeric, and Pi.

Referenced by EllipticK_eval(), and REGISTER_FUNCTION().

7.1.3.306 EllipticK_deriv()

```
ex GiNaC::EllipticK_deriv (
            const ex & k,
            unsigned deriv_param)  [static]
```

References EllipticK_deriv().

Referenced by EllipticK_deriv(), and REGISTER_FUNCTION().

7.1.3.307 EllipticK_series()

```
ex GiNaC::EllipticK_series (
            const ex & k,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex0, _ex1, _ex_1, EllipticK_series(), GiNaC::expair::expair(), GiNaC::subs_options::no_pattern, GiNaC::numeric::numeric(), Pi, pow(), GiNaC::pseries::pseries(), GiNaC::ex::series(), and GiNaC::ex::subs().

Referenced by EllipticK_series(), and REGISTER_FUNCTION().

7.1.3.308 EllipticK_print_latex()

```
void GiNaC::EllipticK_print_latex (
            const ex & k,
            const print_context & c)  [static]
```

References EllipticK_print_latex(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by EllipticK_print_latex(), and REGISTER_FUNCTION().

7.1.3.309 REGISTER_FUNCTION() [14/36]

```
GiNaC::REGISTER_FUNCTION (
            EllipticK ,
            evalf_func(EllipticK_evalf).  eval_func(EllipticK_eval).  derivative_func(EllipticK_deriv).
series_func(EllipticK_series).  print_func< print_latex >(EllipticK_print_latex).  do_not_↩
evalf_params() )
```

References EllipticK_deriv(), EllipticK_eval(), EllipticK_evalf(), EllipticK_print_latex(), EllipticK_series(), and REGISTER_FUNCTION().

7.1.3.310 EllipticE_evalf()

```
ex GiNaC::EllipticE_evalf (
             const ex & k)  [static]
```

References EllipticE_evalf(), GiNaC::ex::evalf(), ex_to(), GiNaC::ex::info(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), Pi, and sqrt().

Referenced by EllipticE_evalf(), and REGISTER_FUNCTION().

7.1.3.311 EllipticE_eval()

```
ex GiNaC::EllipticE_eval (
             const ex & k)  [static]
```

References _ex0, _ex1, _ex_1, GiNaC::info_flags::crational, EllipticE_eval(), GiNaC::ex::info(), GiNaC::info_flags::numeric, and Pi.

Referenced by EllipticE_eval(), and REGISTER_FUNCTION().

7.1.3.312 EllipticE_deriv()

```
ex GiNaC::EllipticE_deriv (
             const ex & k,
             unsigned deriv_param)  [static]
```

References EllipticE_deriv().

Referenced by EllipticE_deriv(), and REGISTER_FUNCTION().

7.1.3.313 EllipticE_series()

```
ex GiNaC::EllipticE_series (
             const ex & k,
             const relational & rel,
             int order,
             unsigned options)  [static]
```

References _ex0, _ex1, _ex_1, EllipticE_series(), GiNaC::expair::expair(), GiNaC::subs_options::no_pattern, GiNaC::numeric::numeric(), Pi, pow(), GiNaC::pseries::pseries(), GiNaC::ex::series(), and GiNaC::ex::subs().

Referenced by EllipticE_series(), and REGISTER_FUNCTION().

7.1.3.314 EllipticE_print_latex()

```
void GiNaC::EllipticE_print_latex (
             const ex & k,
             const print_context & c)  [static]
```

References EllipticE_print_latex(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by EllipticE_print_latex(), and REGISTER_FUNCTION().

7.1.3.315 REGISTER_FUNCTION() [15/36]

```
GiNaC::REGISTER_FUNCTION (
            EllipticE ,
            evalf_func(EllipticE_evalf).  eval_func(EllipticE_eval).  derivative_func(EllipticE_deriv).
series_func(EllipticE_series).  print_func< print_latex >(EllipticE_print_latex).  do_not_↩
evalf_params() )
```

References EllipticE_deriv(), EllipticE_eval(), EllipticE_evalf(), EllipticE_print_latex(), EllipticE_series(), and REGISTER_FUNCTION().

7.1.3.316 iterated_integral_evalf_impl()

```
ex GiNaC::iterated_integral_evalf_impl (
            const ex & kernel_lst,
            const ex & lambda,
            const ex & N_trunc)  [static]
```

References coeff(), Digits, GiNaC::ex::evalf(), ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), is_a(), iterated_integral(), iterated_integral_evalf_impl(), GiNaC::info_flags::list, GiNaC::info_flags::nonnegint, GiNaC::container< C >::nops(), GiNaC::info_flags::numeric, and GiNaC::numeric::numeric().

Referenced by iterated_integral2_evalf(), iterated_integral3_evalf(), and iterated_integral_evalf_impl().

7.1.3.317 iterated_integral2_evalf()

```
ex GiNaC::iterated_integral2_evalf (
            const ex & kernel_lst,
            const ex & lambda)  [static]
```

References iterated_integral2_evalf(), and iterated_integral_evalf_impl().

Referenced by iterated_integral2_evalf().

7.1.3.318 iterated_integral3_evalf()

```
ex GiNaC::iterated_integral3_evalf (
            const ex & kernel_lst,
            const ex & lambda,
            const ex & N_trunc)  [static]
```

References iterated_integral3_evalf(), and iterated_integral_evalf_impl().

Referenced by iterated_integral3_evalf().

7.1.3.319 iterated_integral2_eval()

```
ex GiNaC::iterated_integral2_eval (
            const ex & kernel_lst,
            const ex & lambda)  [static]
```

References GiNaC::info_flags::crational, GiNaC::function::evalf(), GiNaC::basic::hold(), GiNaC::ex::info(), iterated_integral(), iterated_integral2_eval(), and GiNaC::info_flags::numeric.

Referenced by iterated_integral2_eval().

7.1.3.320 iterated_integral3_eval()

```
ex GiNaC::iterated_integral3_eval (
            const ex & kernel_lst,
            const ex & lambda,
            const ex & N_trunc)  [static]
```

References GiNaC::info_flags::crational, GiNaC::function::evalf(), GiNaC::basic::hold(), GiNaC::ex::info(), iterated_integral(), iterated_integral3_eval(), and GiNaC::info_flags::numeric.

Referenced by iterated_integral3_eval().

7.1.3.321 lgamma_evalf()

```
ex GiNaC::lgamma_evalf (
            const ex & x)  [static]
```

References ex_to(), is_exactly_a(), lgamma(), and lgamma_evalf().

Referenced by lgamma_evalf(), and REGISTER_FUNCTION().

7.1.3.322 lgamma_eval()

```
ex GiNaC::lgamma_eval (
            const ex & x)  [static]
```

Evaluation of lgamma(x), the natural logarithm of the Gamma function.

Handles integer arguments as a special case.

Exceptions

GiNaC::pole_error("lgamma_eval()	logarithmic pole",0)

References _ex_1, ex_to(), factorial(), GiNaC::ex::info(), GiNaC::info_flags::integer, is_rational(), lgamma(), lgamma_eval(), log(), GiNaC::info_flags::numeric, GiNaC::pole_error::pole_error(), and GiNaC::info_flags::posint.

Referenced by lgamma_eval(), and REGISTER_FUNCTION().

7.1.3.323 lgamma_deriv()

```
ex GiNaC::lgamma_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References GINAC_ASSERT, lgamma_deriv(), and psi().

Referenced by lgamma_deriv(), and REGISTER_FUNCTION().

7.1.3.324 lgamma_series()

```
ex GiNaC::lgamma_series (
             const ex & arg,
             const relational & rel,
             int order,
             unsigned options)  [static]
```

References _ex1, ex_to(), GiNaC::ex::info(), GiNaC::info_flags::integer, lgamma(), lgamma_series(), log(), GiNaC::subs_options::no_pattern, GiNaC::info_flags::positive, GiNaC::basic::series(), and GiNaC::ex::subs().

Referenced by lgamma_series(), and REGISTER_FUNCTION().

7.1.3.325 lgamma_conjugate()

```
ex GiNaC::lgamma_conjugate (
             const ex & x)  [static]
```

References GiNaC::ex::conjugate(), GiNaC::ex::imag_part(), GiNaC::ex::info(), is_exactly_a(), GiNaC::ex::is_zero(), lgamma(), lgamma_conjugate(), and GiNaC::info_flags::positive.

Referenced by lgamma_conjugate(), and REGISTER_FUNCTION().

7.1.3.326 REGISTER_FUNCTION() [16/36]

```
GiNaC::REGISTER_FUNCTION (
             lgamma ,
             eval_func(lgamma_eval).  evalf_func(lgamma_evalf).  derivative_func(lgamma_deriv).
series_func(lgamma_series).  conjugate_func(lgamma_conjugate).  latex_name("\\log \\Gamma") )
```

References lgamma(), lgamma_conjugate(), lgamma_deriv(), lgamma_eval(), lgamma_evalf(), lgamma_series(), and REGISTER_FUNCTION().

7.1.3.327 tgamma_evalf()

```
ex GiNaC::tgamma_evalf (
             const ex & x)  [static]
```

References ex_to(), is_exactly_a(), tgamma(), and tgamma_evalf().

Referenced by REGISTER_FUNCTION(), and tgamma_evalf().

7.1.3.328 tgamma_eval()

```
ex GiNaC::tgamma_eval (
            const ex & x)  [static]
```

Evaluation of tgamma(x), the true Gamma function.

Knows about integer arguments, half-integer arguments and that's it. Somebody ought to provide some good numerical evaluation some day...

Exceptions

pole_error("tgamma_eval()	simple pole",0)

References _num1_2_p, _num1_p, _num2_p, _num_2_p, abs(), GiNaC::numeric::div(), doublefactorial(), ex_to(), factorial(), GiNaC::ex::info(), GiNaC::numeric::is_even(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_positive(), is_rational(), GiNaC::numeric::mul(), GiNaC::info_flags::numeric, Pi, GiNaC::pole_error::pole_error(), pow(), sqrt(), GiNaC::numeric::sub(), tgamma(), and tgamma_eval().

Referenced by REGISTER_FUNCTION(), and tgamma_eval().

7.1.3.329 tgamma_deriv()

```
ex GiNaC::tgamma_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References GINAC_ASSERT, psi(), tgamma(), and tgamma_deriv().

Referenced by REGISTER_FUNCTION(), and tgamma_deriv().

7.1.3.330 tgamma_series()

```
ex GiNaC::tgamma_series (
            const ex & arg,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex1, ex_to(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::subs_options::no_pattern, GiNaC::info_flags::positive, GiNaC::ex::series(), GiNaC::ex::subs(), tgamma(), and tgamma_series().

Referenced by REGISTER_FUNCTION(), and tgamma_series().

7.1.3.331 tgamma_conjugate()

```
ex GiNaC::tgamma_conjugate (
            const ex & x)  [static]
```

References GiNaC::ex::conjugate(), tgamma(), and tgamma_conjugate().

Referenced by REGISTER_FUNCTION(), and tgamma_conjugate().

7.1.3.332 REGISTER_FUNCTION() [17/36]

```
GiNaC::REGISTER_FUNCTION (
             tgamma ,
             eval_func(tgamma_eval).  evalf_func(tgamma_evalf).  derivative_func(tgamma_deriv).
series_func(tgamma_series).  conjugate_func(tgamma_conjugate).  latex_name("\\Gamma") )
```

References REGISTER_FUNCTION(), tgamma(), tgamma_conjugate(), tgamma_deriv(), tgamma_eval(), tgamma_evalf(), and tgamma_series().

7.1.3.333 beta_evalf()

```
ex GiNaC::beta_evalf (
             const ex & x,
             const ex & y)  [static]
```

References beta_evalf(), ex_to(), exp(), is_exactly_a(), and lgamma().

Referenced by beta_evalf(), and REGISTER_FUNCTION().

7.1.3.334 beta_eval()

```
ex GiNaC::beta_eval (
             const ex & x,
             const ex & y)  [static]
```

References _ex0, _ex1, _num_1_p, beta_eval(), evalf(), ex_to(), GiNaC::ex::info(), GiNaC::ex::is_equal(), is_integer(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_negative(), is_positive(), is_rational(), is_real(), GiNaC::numeric::is_real(), GiNaC::info_flags::numeric, GiNaC::pole_error::pole_error(), pow(), and tgamma().

Referenced by beta_eval(), and REGISTER_FUNCTION().

7.1.3.335 beta_deriv()

```
ex GiNaC::beta_deriv (
             const ex & x,
             const ex & y,
             unsigned deriv_param)  [static]
```

References beta_deriv(), GINAC_ASSERT, and psi().

Referenced by beta_deriv(), and REGISTER_FUNCTION().

7.1.3.336 beta_series()

```
ex GiNaC::beta_series (
             const ex & arg1,
             const ex & arg2,
             const relational & rel,
             int order,
             unsigned options)  [static]
```

References beta_series(), ex_to(), GINAC_ASSERT, GiNaC::ex::info(), GiNaC::info_flags::integer, is_a(), GiNaC::relational::lhs(), GiNaC::subs_options::no_pattern, GiNaC::info_flags::positive, GiNaC::ex::subs(), and tgamma().

Referenced by beta_series(), and REGISTER_FUNCTION().

7.1.3.337 REGISTER_FUNCTION() [18/36]

```
GiNaC::REGISTER_FUNCTION (
            beta ,
            eval_func(beta_eval).  evalf_func(beta_evalf).  derivative_func(beta_deriv).
series_func(beta_series).  latex_name("\\mathrm{B}").  set_symmetry(sy_symm(0, 1)) )
```

References beta_deriv(), beta_eval(), beta_evalf(), beta_series(), REGISTER_FUNCTION(), and sy_symm().

7.1.3.338 psi1_evalf()

```
ex GiNaC::psi1_evalf (
            const ex & x)  [static]
```

References ex_to(), GiNaC::basic::hold(), is_exactly_a(), psi(), and psi1_evalf().

Referenced by psi1_evalf().

7.1.3.339 psi1_eval()

```
ex GiNaC::psi1_eval (
            const ex & x)  [static]
```

Evaluation of digamma-function psi(x).

Somebody ought to provide some good numerical evaluation some day...

References _ex1_2, _ex2, _num2_p, _num_1_p, Euler, ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::numeric::inverse(), is_integer(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_positive(), log(), GiNaC::info_flags::numeric, GiNaC::pole_error::pole_error(), pow(), psi(), and psi1_eval().

Referenced by psi1_eval().

7.1.3.340 psi1_deriv()

```
ex GiNaC::psi1_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex1, GINAC_ASSERT, psi(), and psi1_deriv().

Referenced by psi1_deriv().

7.1.3.341 psi1_series()

```
ex GiNaC::psi1_series (
            const ex & arg,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex1, _ex_1, ex_to(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::subs_options::no_pattern, GiNaC::info_flags::positive, GiNaC::power::power(), psi(), psi1_series(), GiNaC::ex::series(), and GiNaC::ex::subs().

Referenced by psi1_series().

7.1.3.342 psi2_evalf()

```
ex GiNaC::psi2_evalf (
             const ex & n,
             const ex & x)  [static]
```

References ex_to(), GiNaC::basic::hold(), is_exactly_a(), psi(), and psi2_evalf().

Referenced by psi2_evalf().

7.1.3.343 psi2_eval()

```
ex GiNaC::psi2_eval (
             const ex & n,
             const ex & x)  [static]
```

Evaluation of polygamma-function psi(n,x).

Somebody ought to provide some good numerical evaluation some day...

References _ex1, _ex1_2, _ex_1, _num1_2_p, _num1_p, _num2_p, _num_1_p, GiNaC::ex::ex(), ex_to(), factorial(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::ex::is_equal(), GiNaC::numeric::is_equal(), is_integer(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_positive(), GiNaC::ex::is_zero(), log(), GiNaC::info_flags::numeric, GiNaC::pole_error::pole_error(), GiNaC::info_flags::posint, pow(), psi(), psi2_eval(), tgamma(), and zeta().

Referenced by psi2_eval().

7.1.3.344 psi2_deriv()

```
ex GiNaC::psi2_deriv (
             const ex & n,
             const ex & x,
             unsigned deriv_param)  [static]
```

References _ex1, GINAC_ASSERT, psi(), and psi2_deriv().

Referenced by psi2_deriv().

7.1.3.345 psi2_series()

```
ex GiNaC::psi2_series (
             const ex & n,
             const ex & arg,
             const relational & rel,
             int order,
             unsigned options)  [static]
```

References _ex1, _ex_1, ex_to(), factorial(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::subs_options::no_pattern, GiNaC::info_flags::positive, GiNaC::power::power(), psi(), psi2_series(), and GiNaC::ex::subs().

Referenced by psi2_series().

7.1.3.346 G2_evalf()

```
ex GiNaC::G2_evalf (
             const ex & x_,
             const ex & y)  [static]
```

References _ex0, _ex1, ex_to(), factorial(), G(), G2_evalf(), GiNaC::basic::hold(), imag(), GiNaC::ex::info(), is_a(), is_real(), log(), GiNaC::container< C >::nops(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), GiNaC::container< C >::op(), GiNaC::info_flags::positive, and pow().

Referenced by G2_evalf().

7.1.3.347 G2_eval()

```
ex GiNaC::G2_eval (
             const ex & x_,
             const ex & y)  [static]
```

References _ex0, _ex1, GiNaC::info_flags::crational, ex_to(), factorial(), G(), G2_eval(), GiNaC::basic::hold(), imag(), GiNaC::ex::info(), is_a(), is_real(), log(), GiNaC::container< C >::nops(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), GiNaC::container< C >::op(), GiNaC::info_flags::positive, and pow().

Referenced by G2_eval().

7.1.3.348 G3_evalf()

```
ex GiNaC::G3_evalf (
             const ex & x_,
             const ex & s_,
             const ex & y)  [static]
```

References _ex0, _ex1, GiNaC::container< C >::begin(), GiNaC::container< C >::end(), ex_to(), factorial(), G(), G3_evalf(), GiNaC::basic::hold(), GiNaC::ex::info(), is_a(), log(), GiNaC::container< C >::nops(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), GiNaC::container< C >::op(), GiNaC::info_flags::positive, pow(), and GiNaC::info_flags::real.

Referenced by G3_evalf().

7.1.3.349 G3_eval()

```
ex GiNaC::G3_eval (
             const ex & x_,
             const ex & s_,
             const ex & y)  [static]
```

References _ex0, _ex1, GiNaC::container< C >::begin(), GiNaC::info_flags::crational, GiNaC::container< C >::end(), ex_to(), factorial(), G(), G3_eval(), GiNaC::basic::hold(), GiNaC::ex::info(), is_a(), log(), GiNaC::container< C >::nops(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), GiNaC::container< C >::op(), GiNaC::info_flags::positive, pow(), and GiNaC::info_flags::real.

Referenced by G3_eval().

7.1.3.350 Li_evalf()

```
ex GiNaC::Li_evalf (
            const ex & m_,
            const ex & x_)  [static]
```

References _ex0, _ex1, GiNaC::container< C >::begin(), GiNaC::container< C >::end(), GiNaC::ex::evalf(), ex_to(), GiNaC::ex::info(), is_a(), Li_evalf(), GiNaC::container< C >::nops(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), GiNaC::container< C >::op(), and GiNaC::info_flags::posint.

Referenced by Li_evalf(), and REGISTER_FUNCTION().

7.1.3.351 Li_eval()

```
ex GiNaC::Li_eval (
            const ex & m_,
            const ex & x_)  [static]
```

References _ex0, _ex1, _ex2, _ex_1, _ex_48, GiNaC::container< C >::append(), GiNaC::container< C >::begin(), Catalan, GiNaC::info_flags::crational, GiNaC::container< C >::end(), ex_to(), GINAC_ASSERT, I, GiNaC::ex::info(), is_a(), GiNaC::ex::is_equal(), Li_eval(), log(), GiNaC::container< C >::nops(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), Pi, GiNaC::info_flags::posint, pow(), GiNaC::power::power(), and zeta().

Referenced by Li_eval(), and REGISTER_FUNCTION().

7.1.3.352 Li_series()

```
ex GiNaC::Li_series (
            const ex & m,
            const ex & x,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex1, GiNaC::expair::expair(), GiNaC::ex::info(), is_a(), GiNaC::ex::is_zero(), Li_series(), GiNaC::subs_options::no_patter GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), pow(), GiNaC::pseries::pseries(), GiNaC::ex::series(), and GiNaC::ex::subs().

Referenced by Li_series(), and REGISTER_FUNCTION().

7.1.3.353 Li_deriv()

```
ex GiNaC::Li_deriv (
            const ex & m_,
            const ex & x_,
            unsigned deriv_param)  [static]
```

References _ex0, GINAC_ASSERT, is_a(), Li_deriv(), GiNaC::ex::nops(), and GiNaC::ex::op().

Referenced by Li_deriv(), and REGISTER_FUNCTION().

7.1.3.354 Li_print_latex()

```
void GiNaC::Li_print_latex (
             const ex & m_,
             const ex & x_,
             const print_context & c)  [static]
```

References GiNaC::container< C >::begin(), GiNaC::container< C >::end(), ex_to(), is_a(), Li_print_latex(), and GiNaC::print_context::s.

Referenced by Li_print_latex(), and REGISTER_FUNCTION().

7.1.3.355 REGISTER_FUNCTION() [19/36]

```
GiNaC::REGISTER_FUNCTION (
             Li ,
             evalf_func(Li_evalf).  eval_func(Li_eval).  series_func(Li_series).  derivative_↩
func(Li_deriv).  print_func< print_latex >(Li_print_latex).  do_not_evalf_params() )
```

References Li_deriv(), Li_eval(), Li_evalf(), Li_print_latex(), Li_series(), and REGISTER_FUNCTION().

7.1.3.356 S_evalf()

```
ex GiNaC::S_evalf (
             const ex & n,
             const ex & p,
             const ex & x)  [static]
```

References GiNaC::ex::evalf(), ex_to(), GiNaC::ex::info(), is_a(), GiNaC::numeric::numeric(), GiNaC::info_flags::posint, and S_evalf().

Referenced by REGISTER_FUNCTION(), and S_evalf().

7.1.3.357 S_eval()

```
ex GiNaC::S_eval (
             const ex & n,
             const ex & p,
             const ex & x)  [static]
```

References _ex0, GiNaC::container< C >::append(), GiNaC::info_flags::crational, ex_to(), factorial(), GiNaC::ex::info(), GiNaC::ex::is_zero(), log(), GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), GiNaC::info_flags::posint, pow(), S_eval(), to_int(), and zeta().

Referenced by REGISTER_FUNCTION(), and S_eval().

7.1.3.358 S_series()

```
ex GiNaC::S_series (
              const ex & n,
              const ex & p,
              const ex & x,
              const relational & rel,
              int order,
              unsigned options)  [static]
```

References _ex1, GiNaC::expair::expair(), GiNaC::ex::info(), GiNaC::ex::is_zero(), GiNaC::subs_options::no_pattern, GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), GiNaC::info_flags::posint, pow(), GiNaC::pseries::pseries(), S_series(), GiNaC::ex::series(), and GiNaC::ex::subs().

Referenced by REGISTER_FUNCTION(), and S_series().

7.1.3.359 S_deriv()

```
ex GiNaC::S_deriv (
              const ex & n,
              const ex & p,
              const ex & x,
              unsigned deriv_param)  [static]
```

References _ex0, GINAC_ASSERT, and S_deriv().

Referenced by REGISTER_FUNCTION(), and S_deriv().

7.1.3.360 S_print_latex()

```
void GiNaC::S_print_latex (
              const ex & n,
              const ex & p,
              const ex & x,
              const print_context & c)  [static]
```

References GiNaC::ex::print(), GiNaC::print_context::s, and S_print_latex().

Referenced by REGISTER_FUNCTION(), and S_print_latex().

7.1.3.361 REGISTER_FUNCTION() [20/36]

```
GiNaC::REGISTER_FUNCTION (
              S ,
              evalf_func(S_evalf).  eval_func(S_eval).  series_func(S_series).  derivative_↩
func(S_deriv).  print_func< print_latex >(S_print_latex).  do_not_evalf_params() )
```

References REGISTER_FUNCTION(), S_deriv(), S_eval(), S_evalf(), S_print_latex(), and S_series().

7.1.3.362 H_evalf()

```
ex GiNaC::H_evalf (
            const ex & x1,
            const ex & x2)  [static]
```

References GiNaC::container< C >::append(), GiNaC::container< C >::begin(), GiNaC::container< C >::end(), GiNaC::ex::evalf(), ex_to(), H_evalf(), I, GiNaC::ex::info(), GiNaC::info_flags::integer, is_a(), GiNaC::container< C >::let_op(), GiNaC::container< C >::nops(), GiNaC::ex::nops(), GiNaC::numeric::numeric(), GiNaC::container< C >::op(), GiNaC::ex::op(), Pi, GiNaC::ex::subs(), and to_int().

Referenced by H_evalf(), and REGISTER_FUNCTION().

7.1.3.363 H_eval()

```
ex GiNaC::H_eval (
            const ex & m_,
            const ex & x)  [static]
```

References _ex0, _ex1, _ex_1, GiNaC::container< C >::begin(), GiNaC::info_flags::crational, GiNaC::container< C >::end(), ex_to(), factorial(), H_eval(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::info_flags::integer, is_a(), log(), GiNaC::container< C >::nops(), GiNaC::info_flags::numeric, pow(), and step().

Referenced by H_eval(), and REGISTER_FUNCTION().

7.1.3.364 H_series()

```
ex GiNaC::H_series (
            const ex & m,
            const ex & x,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References GiNaC::expair::expair(), H_series(), and GiNaC::pseries::pseries().

Referenced by H_series(), and REGISTER_FUNCTION().

7.1.3.365 H_deriv()

```
ex GiNaC::H_deriv (
            const ex & m_,
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex0, _ex1, _ex_1, GiNaC::container< C >::begin(), ex_to(), GINAC_ASSERT, H_deriv(), is_a(), and GiNaC::container< C >::remove_first().

Referenced by H_deriv(), and REGISTER_FUNCTION().

7.1.3.366 H_print_latex()

```
void GiNaC::H_print_latex (
             const ex & m_,
             const ex & x,
             const print_context & c)  [static]
```

References GiNaC::container< C >::begin(), GiNaC::container< C >::end(), ex_to(), H_print_latex(), is_a(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by H_print_latex(), and REGISTER_FUNCTION().

7.1.3.367 REGISTER_FUNCTION() [21/36]

```
GiNaC::REGISTER_FUNCTION (
             H ,
             evalf_func(H_evalf).  eval_func(H_eval).  series_func(H_series).  derivative_↩
func(H_deriv).  print_func< print_latex >(H_print_latex).  do_not_evalf_params() )
```

References H_deriv(), H_eval(), H_evalf(), H_print_latex(), H_series(), and REGISTER_FUNCTION().

7.1.3.368 zeta1_evalf()

```
ex GiNaC::zeta1_evalf (
             const ex & x)  [static]
```

References GiNaC::container< C >::begin(), Digits, ex_to(), GiNaC::basic::hold(), is_exactly_a(), GiNaC::ex::nops(), GiNaC::numeric::numeric(), GiNaC::info_flags::posint, zeta(), and zeta1_evalf().

Referenced by zeta1_eval(), and zeta1_evalf().

7.1.3.369 zeta1_eval()

```
ex GiNaC::zeta1_eval (
             const ex & m)  [static]
```

References _ex0, _ex_1_2, _num1_p, _num2_p, abs(), bernoulli(), GiNaC::info_flags::crational, ex_to(), factorial(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::numeric::info(), GiNaC::numeric::is_equal(), is_exactly_a(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_zero(), GiNaC::ex::nops(), GiNaC::info_flags::numeric, GiNaC::info_flags::odd, GiNaC::ex::op(), Pi, GiNaC::info_flags::posint, pow(), zeta(), zeta1_eval(), and zeta1_evalf().

Referenced by zeta1_eval().

7.1.3.370 zeta1_deriv()

```
ex GiNaC::zeta1_deriv (
             const ex & m,
             unsigned deriv_param)  [static]
```

References _ex0, _ex1, GINAC_ASSERT, is_exactly_a(), and zeta1_deriv().

Referenced by zeta1_deriv().

7.1.3.371 zeta1_print_latex()

```
void GiNaC::zeta1_print_latex (
            const ex & m_,
            const print_context & c)  [static]
```

References GiNaC::container< C >::begin(), GiNaC::container< C >::end(), ex_to(), is_a(), GiNaC::ex::print(), GiNaC::print_context::s, and zeta1_print_latex().

Referenced by zeta1_print_latex().

7.1.3.372 zeta2_evalf()

```
ex GiNaC::zeta2_evalf (
            const ex & x,
            const ex & s)  [static]
```

References GiNaC::container< C >::begin(), evalf(), ex_to(), GiNaC::basic::hold(), is_exactly_a(), GiNaC::ex::nops(), GiNaC::numeric::numeric(), GiNaC::info_flags::posint, zeta(), and zeta2_evalf().

Referenced by zeta2_evalf().

7.1.3.373 zeta2_eval()

```
ex GiNaC::zeta2_eval (
            const ex & m,
            const ex & s_)  [static]
```

References ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), is_exactly_a(), GiNaC::info_flags::positive, zeta(), and zeta2_eval().

Referenced by zeta2_eval().

7.1.3.374 zeta2_deriv()

```
ex GiNaC::zeta2_deriv (
            const ex & m,
            const ex & s,
            unsigned deriv_param)  [static]
```

References _ex0, _ex1, GINAC_ASSERT, GiNaC::ex::info(), is_exactly_a(), GiNaC::ex::op(), GiNaC::info_flags::positive, and zeta2_deriv().

Referenced by zeta2_deriv().

7.1.3.375 zeta2_print_latex()

```
void GiNaC::zeta2_print_latex (
            const ex & m_,
            const ex & s_,
            const print_context & c)  [static]
```

References GiNaC::container< C >::begin(), GiNaC::container< C >::end(), ex_to(), is_a(), GiNaC::print_context::s, and zeta2_print_latex().

Referenced by zeta2_print_latex().

7.1.3.376 exp_evalf()

```
ex GiNaC::exp_evalf (
            const ex & x)  [static]
```

References ex_to(), exp(), exp_evalf(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by exp_evalf(), and REGISTER_FUNCTION().

7.1.3.377 exp_eval()

```
ex GiNaC::exp_eval (
            const ex & x)  [static]
```

References _ex1, _ex2, _ex_1, _num0_p, _num1_p, _num2_p, _num3_p, _num4_p, GiNaC::info_flags::crational, GiNaC::ex::ex(), ex_to(), exp(), exp_eval(), GiNaC::basic::hold(), I, GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::numeric::is_equal(), is_ex_the_function, GiNaC::ex::is_zero(), log(), mod(), GiNaC::info_flags::numeric, GiNaC::ex::op(), and Pi.

Referenced by exp_eval(), and REGISTER_FUNCTION().

7.1.3.378 exp_expand()

```
ex GiNaC::exp_expand (
            const ex & arg,
            unsigned options)  [static]
```

References GiNaC::ex::begin(), dynallocate(), GiNaC::ex::end(), exp(), exp_expand(), GiNaC::ex::expand(), GiNaC::expand_options::expand_function_args, GiNaC::expand_options::expand_transcendental, GiNaC::status_flags::expanded, GiNaC::basic::hold(), is_exactly_a(), and GiNaC::ex::nops().

Referenced by exp_expand(), and REGISTER_FUNCTION().

7.1.3.379 exp_deriv()

```
ex GiNaC::exp_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References exp(), exp_deriv(), and GINAC_ASSERT.

Referenced by exp_deriv(), and REGISTER_FUNCTION().

7.1.3.380 exp_real_part()

```
ex GiNaC::exp_real_part (
            const ex & x)  [static]
```

References cos(), exp(), exp_real_part(), imag_part(), and real_part().

Referenced by exp_real_part(), and REGISTER_FUNCTION().

7.1.3.381 exp_imag_part()

```
ex GiNaC::exp_imag_part (
            const ex & x)  [static]
```

References exp(), exp_imag_part(), imag_part(), real_part(), and sin().

Referenced by exp_imag_part(), and REGISTER_FUNCTION().

7.1.3.382 exp_conjugate()

```
ex GiNaC::exp_conjugate (
            const ex & x)  [static]
```

References GiNaC::ex::conjugate(), exp(), and exp_conjugate().

Referenced by exp_conjugate(), and REGISTER_FUNCTION().

7.1.3.383 exp_power()

```
ex GiNaC::exp_power (
            const ex & x,
            const ex & a)  [static]
```

References _ex_1, exp(), exp_power(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::info_flags::negative, GiNaC::info_flags::nonnegative, GiNaC::power::power(), and GiNaC::info_flags::real.

Referenced by exp_power(), and REGISTER_FUNCTION().

7.1.3.384 exp_info()

```
bool GiNaC::exp_info (
            const ex & x,
            unsigned inf)  [static]
```

References exp_info(), GiNaC::info_flags::expanded, GiNaC::ex::info(), GiNaC::info_flags::nonnegative, GiNaC::info_flags::positive, and GiNaC::info_flags::real.

Referenced by exp_info(), REGISTER_FUNCTION(), and REGISTER_FUNCTION().

7.1.3.385 REGISTER_FUNCTION() [22/36]

```
GiNaC::REGISTER_FUNCTION (
            exp ,
            eval_func(exp_eval).  evalf_func(exp_evalf).  info_func(exp_info).  expand_↩
func(exp_expand).  derivative_func(exp_deriv).  real_part_func(exp_real_part).  imag_part_↩
func(exp_imag_part).  conjugate_func(exp_conjugate).  power_func(exp_power).  latex_name("\\exp")
)
```

References exp(), exp_conjugate(), exp_deriv(), exp_eval(), exp_evalf(), exp_expand(), exp_imag_part(), exp_info(), exp_power(), exp_real_part(), and REGISTER_FUNCTION().

7.1.3.386 log_evalf()

```
ex GiNaC::log_evalf (
            const ex & x)  [static]
```

References ex_to(), GiNaC::basic::hold(), is_exactly_a(), log(), and log_evalf().

Referenced by log_evalf(), and REGISTER_FUNCTION().

7.1.3.387 log_eval()

```
ex GiNaC::log_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex1_2, _ex_1_2, GiNaC::info_flags::crational, ex_to(), exp(), GiNaC::basic::hold(), I, GiNaC::ex::info(), GiNaC::ex::is_equal(), is_ex_the_function, GiNaC::ex::is_zero(), log(), log_eval(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::ex::op(), Pi, GiNaC::pole_error::pole_error(), GiNaC::info_flags::rational, and GiNaC::info_flags::real.

Referenced by log_eval(), and REGISTER_FUNCTION().

7.1.3.388 log_deriv()

```
ex GiNaC::log_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex_1, GINAC_ASSERT, log_deriv(), and GiNaC::power::power().

Referenced by log_deriv(), and REGISTER_FUNCTION().

7.1.3.389 log_series()

```
ex GiNaC::log_series (
            const ex & arg,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex0, _ex1, _ex_1, coeff(), GiNaC::pseries::coeff(), csgn(), GiNaC::ex::diff(), dynallocate(), ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, I, GiNaC::ex::info(), is_a(), GiNaC::pseries::is_terminating(), GiNaC::ex::is_zero(), GiNaC::pseries::ldegree(), GiNaC::relational::lhs(), log(), log_series(), GiNaC::info_flags::negative, GiNaC::subs_options::no_pattern, GiNaC::pseries::nops(), Pi, GiNaC::info_flags::positive, pow(), GiNaC::pseries::pseries(), GiNaC::relational::rhs(), GiNaC::ex::series(), series(), GiNaC::ex::subs(), and GiNaC::series_options::suppress_branchcut.

Referenced by log_series(), and REGISTER_FUNCTION().

7.1.3.390 log_real_part()

```
ex GiNaC::log_real_part (
            const ex & x)  [static]
```

References abs(), GiNaC::basic::hold(), GiNaC::ex::info(), log(), log_real_part(), and GiNaC::info_flags::nonnegative.

Referenced by log_real_part(), and REGISTER_FUNCTION().

7.1.3.391 log_imag_part()

```
ex GiNaC::log_imag_part (
            const ex & x)  [static]
```

References imag_part(), GiNaC::ex::info(), log_imag_part(), GiNaC::info_flags::nonnegative, and real_part().

Referenced by log_imag_part(), and REGISTER_FUNCTION().

7.1.3.392 log_expand()

```
ex GiNaC::log_expand (
            const ex & arg,
            unsigned options)  [static]
```

References _ex1, _ex_1, GiNaC::add::add(), GiNaC::ex::begin(), GiNaC::ex::end(), ex_to(), GiNaC::ex::expand(), expand(), GiNaC::expand_options::expand_function_args, GiNaC::expand_options::expand_transcendental, GiNaC::basic::hold(), GiNaC::info_flags::indefinite, GiNaC::ex::info(), is_exactly_a(), log(), log_expand(), GiNaC::mul::mul(), GiNaC::info_flags::negative, GiNaC::ex::nops(), GiNaC::info_flags::positive, and GiNaC::status_flags::purely_indef

Referenced by log_expand(), and REGISTER_FUNCTION().

7.1.3.393 log_conjugate()

```
ex GiNaC::log_conjugate (
            const ex & x)  [static]
```

References GiNaC::ex::conjugate(), GiNaC::ex::imag_part(), GiNaC::ex::info(), is_exactly_a(), GiNaC::ex::is_zero(), log(), log_conjugate(), and GiNaC::info_flags::positive.

Referenced by log_conjugate(), and REGISTER_FUNCTION().

7.1.3.394 log_info()

```
bool GiNaC::log_info (
            const ex & x,
            unsigned inf)  [static]
```

References GiNaC::info_flags::expanded, GiNaC::ex::info(), log_info(), GiNaC::info_flags::positive, and GiNaC::info_flags::real.

Referenced by log_info(), and REGISTER_FUNCTION().

7.1.3.395 REGISTER_FUNCTION() [23/36]

```
GiNaC::REGISTER_FUNCTION (
            log ,
            eval_func(log_eval).  evalf_func(log_evalf).  info_func(log_info).  expand_↩
func(log_expand).  derivative_func(log_deriv).  series_func(log_series).  real_part_func(log_real_part).
imag_part_func(log_imag_part).  conjugate_func(log_conjugate).  latex_name("\\ln") )
```

References log(), log_conjugate(), log_deriv(), log_eval(), log_evalf(), log_expand(), log_imag_part(), log_info(), log_real_part(), log_series(), and REGISTER_FUNCTION().

7.1.3.396 sin_evalf()

```
ex GiNaC::sin_evalf (
            const ex & x)  [static]
```

References ex_to(), GiNaC::basic::hold(), is_exactly_a(), sin(), and sin_evalf().

Referenced by REGISTER_FUNCTION(), and sin_evalf().

7.1.3.397 sin_eval()

```
ex GiNaC::sin_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex1_2, _ex1_3, _ex1_4, _ex2, _ex3, _ex5, _ex6, _ex60, _ex_1, _ex_1_2, _ex_1_3, _ex_1_4, _num0_p, _num10_p, _num120_p, _num15_p, _num18_p, _num20_p, _num25_p, _num30_p, _num5_p, _num60_p, _num6_p, acos(), asin(), atan(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::numeric::is_equal(), is_ex_the_function, is_exactly_a(), mod(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::ex::op(), Pi, GiNaC::power::power(), sin(), sin_eval(), and sqrt().

Referenced by REGISTER_FUNCTION(), and sin_eval().

7.1.3.398 sin_deriv()

```
ex GiNaC::sin_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References cos(), GINAC_ASSERT, and sin_deriv().

Referenced by REGISTER_FUNCTION(), and sin_deriv().

7.1.3.399 sin_real_part()

```
ex GiNaC::sin_real_part (
            const ex & x)  [static]
```

References cosh(), imag_part(), real_part(), sin(), and sin_real_part().

Referenced by REGISTER_FUNCTION(), and sin_real_part().

7.1.3.400 sin_imag_part()

```
ex GiNaC::sin_imag_part (
            const ex & x)  [static]
```

References cos(), imag_part(), real_part(), sin_imag_part(), and sinh().

Referenced by REGISTER_FUNCTION(), and sin_imag_part().

7.1.3.401 sin_conjugate()

```
ex GiNaC::sin_conjugate (
            const ex & x)  [static]
```

References GiNaC::ex::conjugate(), sin(), and sin_conjugate().

Referenced by REGISTER_FUNCTION(), and sin_conjugate().

7.1.3.402 trig_info()

```
bool GiNaC::trig_info (
            const ex & x,
            unsigned inf)  [static]
```

References GiNaC::info_flags::expanded, GiNaC::ex::info(), GiNaC::info_flags::real, and trig_info().

Referenced by REGISTER_FUNCTION(), REGISTER_FUNCTION(), REGISTER_FUNCTION(), and trig_info().

7.1.3.403 REGISTER_FUNCTION() [24/36]

```
GiNaC::REGISTER_FUNCTION (
            sin ,
            eval_func(sin_eval).  evalf_func(sin_evalf).  info_func(trig_info).  derivative_↩
func(sin_deriv).  real_part_func(sin_real_part).  imag_part_func(sin_imag_part).  conjugate_↩
func(sin_conjugate).  latex_name("\\sin") )
```

References REGISTER_FUNCTION(), sin(), sin_conjugate(), sin_deriv(), sin_eval(), sin_evalf(), sin_imag_part(), sin_real_part(), and trig_info().

7.1.3.404 cos_evalf()

```
ex GiNaC::cos_evalf (
            const ex & x)  [static]
```

References cos(), cos_evalf(), ex_to(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by cos_evalf(), and REGISTER_FUNCTION().

7.1.3.405 cos_eval()

```
ex GiNaC::cos_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex1_2, _ex1_3, _ex1_4, _ex2, _ex3, _ex5, _ex6, _ex60, _ex_1, _ex_1_2, _ex_1_3, _ex_1_4, _num0_p, _num10_p, _num120_p, _num12_p, _num15_p, _num20_p, _num24_p, _num25_p, _num30_p, _num5_p, _num60_p, acos(), asin(), atan(), cos(), cos_eval(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::numeric::is_equal(), is_ex_the_function, is_exactly_a(), mod(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::ex::op(), Pi, GiNaC::power::power(), and sqrt().

Referenced by cos_eval(), and REGISTER_FUNCTION().

7.1.3.406 cos_deriv()

```
ex GiNaC::cos_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References cos_deriv(), GINAC_ASSERT, and sin().

Referenced by cos_deriv(), and REGISTER_FUNCTION().

7.1.3.407 cos_real_part()

```
ex GiNaC::cos_real_part (
            const ex & x)  [static]
```

References cos(), cos_real_part(), cosh(), imag_part(), and real_part().

Referenced by cos_real_part(), and REGISTER_FUNCTION().

7.1.3.408 cos_imag_part()

```
ex GiNaC::cos_imag_part (
            const ex & x)  [static]
```

References cos_imag_part(), imag_part(), real_part(), sin(), and sinh().

Referenced by cos_imag_part(), and REGISTER_FUNCTION().

7.1.3.409 cos_conjugate()

```
ex GiNaC::cos_conjugate (
            const ex & x)  [static]
```

References GiNaC::ex::conjugate(), cos(), and cos_conjugate().

Referenced by cos_conjugate(), and REGISTER_FUNCTION().

7.1.3.410 REGISTER_FUNCTION() [25/36]

```
GiNaC::REGISTER_FUNCTION (
            cos ,
            eval_func(cos_eval).  info_func(trig_info).  evalf_func(cos_evalf).  derivative_↩
func(cos_deriv).  real_part_func(cos_real_part).  imag_part_func(cos_imag_part).  conjugate_↩
func(cos_conjugate).  latex_name("\\cos") )
```

References cos(), cos_conjugate(), cos_deriv(), cos_eval(), cos_evalf(), cos_imag_part(), cos_real_part(), REGISTER_FUNCTION(), and trig_info().

7.1.3.411 tan_evalf()

```
ex GiNaC::tan_evalf (
            const ex & x)  [static]
```

References ex_to(), GiNaC::basic::hold(), is_exactly_a(), tan(), and tan_evalf().

Referenced by REGISTER_FUNCTION(), and tan_evalf().

7.1.3.412 tan_eval()

```
ex GiNaC::tan_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex1_3, _ex2, _ex3, _ex60, _ex_1, _ex_1_2, _num0_p, _num10_p, _num15_p, _num20_p, _num25_p, _num30_p, _num5_p, _num60_p, acos(), asin(), atan(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::numeric::is_equal(), is_ex_the_function, is_exactly_a(), mod(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::ex::op(), Pi, GiNaC::pole_error::pole_error(), GiNaC::power::power(), sqrt(), tan(), and tan_eval().

Referenced by REGISTER_FUNCTION(), and tan_eval().

7.1.3.413 tan_deriv()

```
ex GiNaC::tan_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex1, _ex2, GINAC_ASSERT, GiNaC::power::power(), tan(), and tan_deriv().

Referenced by REGISTER_FUNCTION(), and tan_deriv().

7.1.3.414 tan_real_part()

```
ex GiNaC::tan_real_part (
            const ex & x)  [static]
```

References imag_part(), GiNaC::power::power(), real_part(), tan(), and tan_real_part().

Referenced by REGISTER_FUNCTION(), and tan_real_part().

7.1.3.415 tan_imag_part()

```
ex GiNaC::tan_imag_part (
            const ex & x)  [static]
```

References imag_part(), GiNaC::power::power(), real_part(), tan(), tan_imag_part(), and tanh().

Referenced by REGISTER_FUNCTION(), and tan_imag_part().

7.1.3.416 tan_series()

```
ex GiNaC::tan_series (
            const ex & x,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References cos(), GINAC_ASSERT, is_a(), GiNaC::relational::lhs(), GiNaC::subs_options::no_pattern, GiNaC::info_flags::odd, Pi, sin(), GiNaC::ex::subs(), and tan_series().

Referenced by REGISTER_FUNCTION(), and tan_series().

7.1.3.417 tan_conjugate()

```
ex GiNaC::tan_conjugate (
            const ex & x)  [static]
```

References GiNaC::ex::conjugate(), tan(), and tan_conjugate().

Referenced by REGISTER_FUNCTION(), and tan_conjugate().

7.1.3.418 REGISTER_FUNCTION() [26/36]

```
GiNaC::REGISTER_FUNCTION (
            tan ,
            eval_func(tan_eval).  evalf_func(tan_evalf).  info_func(trig_info).  derivative↩
_func(tan_deriv).  series_func(tan_series).  real_part_func(tan_real_part).  imag_part_↩
func(tan_imag_part).  conjugate_func(tan_conjugate).  latex_name("\\tan") )
```

References REGISTER_FUNCTION(), tan(), tan_conjugate(), tan_deriv(), tan_eval(), tan_evalf(), tan_imag_part(), tan_real_part(), tan_series(), and trig_info().

7.1.3.419 asin_evalf()

```
ex GiNaC::asin_evalf (
            const ex & x)  [static]
```

References asin(), asin_evalf(), ex_to(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by asin_evalf(), and REGISTER_FUNCTION().

7.1.3.420 asin_eval()

```
ex GiNaC::asin_eval (
            const ex & x)  [static]
```

References _ex1, _ex1_2, _ex_1, _ex_1_2, asin(), asin_eval(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), and Pi.

Referenced by asin_eval(), and REGISTER_FUNCTION().

7.1.3.421 asin_deriv()

```
ex GiNaC::asin_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex2, _ex_1_2, asin_deriv(), GINAC_ASSERT, and GiNaC::power::power().

Referenced by asin_deriv(), and REGISTER_FUNCTION().

7.1.3.422 asin_conjugate()

```
ex GiNaC::asin_conjugate (
            const ex & x)  [static]
```

References _num1_p, _num_1_p, asin(), asin_conjugate(), GiNaC::ex::conjugate(), GiNaC::ex::imag_part(), is_exactly_a(), and GiNaC::ex::is_zero().

Referenced by asin_conjugate(), and REGISTER_FUNCTION().

7.1.3.423 asin_info()

```
bool GiNaC::asin_info (
            const ex & x,
            unsigned inf)  [static]
```

References asin_info(), GiNaC::info_flags::expanded, and GiNaC::ex::info().

Referenced by asin_info(), REGISTER_FUNCTION(), REGISTER_FUNCTION(), REGISTER_FUNCTION(), and REGISTER_FUNCTION().

7.1.3.424 REGISTER_FUNCTION() [27/36]

```
GiNaC::REGISTER_FUNCTION (
            asin ,
            eval_func(asin_eval).  evalf_func(asin_evalf).  info_func(asin_info).  derivative↩
_func(asin_deriv).  conjugate_func(asin_conjugate).  latex_name("\\arcsin") )
```

References asin(), asin_conjugate(), asin_deriv(), asin_eval(), asin_evalf(), asin_info(), and REGISTER_FUNCTION().

7.1.3.425 acos_evalf()

```
ex GiNaC::acos_evalf (
            const ex & x)  [static]
```

References acos(), acos_evalf(), ex_to(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by acos_evalf(), and REGISTER_FUNCTION().

7.1.3.426 acos_eval()

```
ex GiNaC::acos_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex1_2, _ex1_3, _ex_1, _ex_1_2, acos(), acos_eval(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), and Pi.

Referenced by acos_eval(), and REGISTER_FUNCTION().

7.1.3.427 acos_deriv()

```
ex GiNaC::acos_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex2, _ex_1_2, acos_deriv(), GINAC_ASSERT, and GiNaC::power::power().

Referenced by acos_deriv(), and REGISTER_FUNCTION().

7.1.3.428 acos_conjugate()

```
ex GiNaC::acos_conjugate (
            const ex & x)  [static]
```

References _num1_p, _num_1_p, acos(), acos_conjugate(), GiNaC::ex::conjugate(), GiNaC::ex::imag_part(), is_exactly_a(), and GiNaC::ex::is_zero().

Referenced by acos_conjugate(), and REGISTER_FUNCTION().

7.1.3.429 REGISTER_FUNCTION() [28/36]

```
GiNaC::REGISTER_FUNCTION (
            acos ,
            eval_func(acos_eval).  evalf_func(acos_evalf).  info_func(asin_info).  derivative↩
_func(acos_deriv).  conjugate_func(acos_conjugate).  latex_name("\\arccos") )
```

References acos(), acos_conjugate(), acos_deriv(), acos_eval(), acos_evalf(), asin_info(), and REGISTER_FUNCTION().

7.1.3.430 atan_evalf()

```
ex GiNaC::atan_evalf (
            const ex & x)  [static]
```

References atan(), atan_evalf(), ex_to(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by atan_evalf(), and REGISTER_FUNCTION().

7.1.3.431 atan_eval()

```
ex GiNaC::atan_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex1_4, _ex_1, _ex_1_4, atan(), atan_eval(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), I, GiNaC::ex::info(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, Pi, and GiNaC::pole_error::pole_error().

Referenced by atan_eval(), and REGISTER_FUNCTION().

7.1.3.432 atan_deriv()

```
ex GiNaC::atan_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex1, _ex2, _ex_1, atan_deriv(), GINAC_ASSERT, and GiNaC::power::power().

Referenced by atan_deriv(), and REGISTER_FUNCTION().

7.1.3.433 atan_series()

```
ex GiNaC::atan_series (
            const ex & arg,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex0, _ex1, _ex1_2, _ex_1, _ex_1_2, abs(), atan(), atan_series(), csgn(), ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, I, is_a(), GiNaC::ex::is_equal(), GiNaC::relational::lhs(), log(), GiNaC::subs_options::no_pattern, GiNaC::ex::op(), Pi, GiNaC::pseries::pseries(), GiNaC::info_flags::real, GiNaC::relational::rhs(), series(), GiNaC::ex::subs(), and GiNaC::series_options::suppress_branchcut.

Referenced by atan_series(), and REGISTER_FUNCTION().

7.1.3.434 atan_conjugate()

```
ex GiNaC::atan_conjugate (
            const ex & x)  [static]
```

References _num1_p, _num_1_p, atan(), atan_conjugate(), GiNaC::ex::conjugate(), ex_to(), GiNaC::ex::imag_part(), GiNaC::ex::info(), is_exactly_a(), GiNaC::numeric::is_zero(), GiNaC::info_flags::real, and GiNaC::ex::real_part().

Referenced by atan_conjugate(), and REGISTER_FUNCTION().

7.1.3.435 atan_info()

```
bool GiNaC::atan_info (
            const ex & x,
            unsigned inf)  [static]
```

References atan_info(), GiNaC::info_flags::expanded, GiNaC::ex::info(), GiNaC::info_flags::negative, GiNaC::info_flags::nonnegative, GiNaC::info_flags::positive, and GiNaC::info_flags::real.

Referenced by atan_info(), REGISTER_FUNCTION(), REGISTER_FUNCTION(), REGISTER_FUNCTION(), and REGISTER_FUNCTION().

7.1.3.436 REGISTER_FUNCTION() [29/36]

```
GiNaC::REGISTER_FUNCTION (
            atan ,
            eval_func(atan_eval).  evalf_func(atan_evalf).  info_func(atan_info).  derivative↩
_func(atan_deriv).  series_func(atan_series).  conjugate_func(atan_conjugate).  latex_name("\\arctan")
)
```

References atan(), atan_conjugate(), atan_deriv(), atan_eval(), atan_evalf(), atan_info(), atan_series(), and REGISTER_FUNCTION().

7.1.3.437 atan2_evalf()

```
ex GiNaC::atan2_evalf (
            const ex & y,
            const ex & x)  [static]
```

References atan(), atan2_evalf(), ex_to(), and is_exactly_a().

Referenced by atan2_evalf(), and REGISTER_FUNCTION().

7.1.3.438 atan2_eval()

```
ex GiNaC::atan2_eval (
            const ex & y,
            const ex & x)  [static]
```

References _ex0, _ex1_2, _ex1_4, _ex_1_2, _ex_1_4, atan(), atan2_eval(), GiNaC::info_flags::crational, ex_to(), GiNaC::ex::info(), is_a(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), GiNaC::info_flags::negative, GiNaC::numeric::numeric(), Pi, GiNaC::info_flags::positive, and GiNaC::info_flags::real.

Referenced by atan2_eval(), and REGISTER_FUNCTION().

7.1.3.439 atan2_deriv()

```
ex GiNaC::atan2_deriv (
            const ex & y,
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex2, _ex_1, atan2_deriv(), GINAC_ASSERT, and GiNaC::power::power().

Referenced by atan2_deriv(), and REGISTER_FUNCTION().

7.1.3.440 atan2_info()

```
bool GiNaC::atan2_info (
            const ex & y,
            const ex & x,
            unsigned inf)  [static]
```

References atan2_info(), GiNaC::info_flags::expanded, GiNaC::ex::info(), GiNaC::info_flags::negative, GiNaC::info_flags::nonnegative, GiNaC::info_flags::positive, and GiNaC::info_flags::real.

Referenced by atan2_info(), and REGISTER_FUNCTION().

7.1.3.441 REGISTER_FUNCTION() [30/36]

```
GiNaC::REGISTER_FUNCTION (
            atan2 ,
            eval_func(atan2_eval).  evalf_func(atan2_evalf).  info_func(atan2_info).  evalf_↩
func(atan2_evalf).  derivative_func(atan2_deriv) )
```

References atan2_deriv(), atan2_eval(), atan2_evalf(), atan2_info(), and REGISTER_FUNCTION().

7.1.3.442 sinh_evalf()

```
ex GiNaC::sinh_evalf (
            const ex & x)  [static]
```

References ex_to(), GiNaC::basic::hold(), is_exactly_a(), sinh(), and sinh_evalf().

Referenced by REGISTER_FUNCTION(), and sinh_evalf().

7.1.3.443 sinh_eval()

```
ex GiNaC::sinh_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex2, _ex_1_2, acosh(), asinh(), atanh(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), I, GiNaC::ex::info(), is_ex_the_function, is_exactly_a(), GiNaC::ex::is_zero(), is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::ex::op(), Pi, GiNaC::power::power(), real(), sin(), sinh(), sinh_eval(), and sqrt().

Referenced by REGISTER_FUNCTION(), and sinh_eval().

7.1.3.444 sinh_deriv()

```
ex GiNaC::sinh_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References cosh(), GINAC_ASSERT, and sinh_deriv().

Referenced by REGISTER_FUNCTION(), and sinh_deriv().

7.1.3.445 sinh_real_part()

```
ex GiNaC::sinh_real_part (
            const ex & x)  [static]
```

References cos(), imag_part(), real_part(), sinh(), and sinh_real_part().

Referenced by REGISTER_FUNCTION(), and sinh_real_part().

7.1.3.446 sinh_imag_part()

```
ex GiNaC::sinh_imag_part (
            const ex & x)  [static]
```

References cosh(), imag_part(), real_part(), sin(), and sinh_imag_part().

Referenced by REGISTER_FUNCTION(), and sinh_imag_part().

7.1.3.447 sinh_conjugate()

```
ex GiNaC::sinh_conjugate (
            const ex & x)  [static]
```

References GiNaC::ex::conjugate(), sinh(), and sinh_conjugate().

Referenced by REGISTER_FUNCTION(), and sinh_conjugate().

7.1.3.448 REGISTER_FUNCTION() [31/36]

```
GiNaC::REGISTER_FUNCTION (
            sinh ,
            eval_func(sinh_eval).  evalf_func(sinh_evalf).  info_func(atan_info).  derivative↩
_func(sinh_deriv).  real_part_func(sinh_real_part).  imag_part_func(sinh_imag_part).  conjugate↩
_func(sinh_conjugate).  latex_name("\\sinh") )
```

References atan_info(), REGISTER_FUNCTION(), sinh(), sinh_conjugate(), sinh_deriv(), sinh_eval(), sinh_evalf(), sinh_imag_part(), and sinh_real_part().

7.1.3.449 cosh_evalf()

```
ex GiNaC::cosh_evalf (
            const ex & x)  [static]
```

References cosh(), cosh_evalf(), ex_to(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by cosh_evalf(), and REGISTER_FUNCTION().

7.1.3.450 cosh_eval()

```
ex GiNaC::cosh_eval (
            const ex & x)  [static]
```

References _ex1, _ex2, _ex_1_2, acosh(), asinh(), atanh(), cos(), cosh(), cosh_eval(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), I, GiNaC::ex::info(), is_ex_the_function, is_exactly_a(), GiNaC::ex::is_zero(), is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::ex::op(), Pi, GiNaC::power::power(), real(), and sqrt().

Referenced by cosh_eval(), and REGISTER_FUNCTION().

7.1.3.451 cosh_deriv()

```
ex GiNaC::cosh_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References cosh_deriv(), GINAC_ASSERT, and sinh().

Referenced by cosh_deriv(), and REGISTER_FUNCTION().

7.1.3.452 cosh_real_part()

```
ex GiNaC::cosh_real_part (
            const ex & x)  [static]
```

References cos(), cosh(), cosh_real_part(), imag_part(), and real_part().

Referenced by cosh_real_part(), and REGISTER_FUNCTION().

7.1.3.453 cosh_imag_part()

```
ex GiNaC::cosh_imag_part (
            const ex & x)  [static]
```

References cosh_imag_part(), imag_part(), real_part(), sin(), and sinh().

Referenced by cosh_imag_part(), and REGISTER_FUNCTION().

7.1.3.454 cosh_conjugate()

```
ex GiNaC::cosh_conjugate (
            const ex & x)  [static]
```

References GiNaC::ex::conjugate(), cosh(), and cosh_conjugate().

Referenced by cosh_conjugate(), and REGISTER_FUNCTION().

7.1.3.455 REGISTER_FUNCTION() [32/36]

```
GiNaC::REGISTER_FUNCTION (
             cosh ,
             eval_func(cosh_eval).  evalf_func(cosh_evalf).  info_func(exp_info).  derivative↩
_func(cosh_deriv).  real_part_func(cosh_real_part).  imag_part_func(cosh_imag_part).  conjugate↩
_func(cosh_conjugate).  latex_name("\\cosh") )
```

References cosh(), cosh_conjugate(), cosh_deriv(), cosh_eval(), cosh_evalf(), cosh_imag_part(), cosh_real_part(), exp_info(), and REGISTER_FUNCTION().

7.1.3.456 tanh_evalf()

```
ex GiNaC::tanh_evalf (
             const ex & x)  [static]
```

References ex_to(), GiNaC::basic::hold(), is_exactly_a(), tanh(), and tanh_evalf().

Referenced by REGISTER_FUNCTION(), and tanh_evalf().

7.1.3.457 tanh_eval()

```
ex GiNaC::tanh_eval (
             const ex & x)  [static]
```

References _ex0, _ex1, _ex2, _ex_1, _ex_1_2, acosh(), asinh(), atanh(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), I, GiNaC::ex::info(), is_ex_the_function, is_exactly_a(), GiNaC::ex::is_zero(), is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::ex::op(), Pi, GiNaC::power::power(), real(), sqrt(), tan(), tanh(), and tanh_eval().

Referenced by REGISTER_FUNCTION(), and tanh_eval().

7.1.3.458 tanh_deriv()

```
ex GiNaC::tanh_deriv (
             const ex & x,
             unsigned deriv_param)  [static]
```

References _ex1, _ex2, GINAC_ASSERT, GiNaC::power::power(), tanh(), and tanh_deriv().

Referenced by REGISTER_FUNCTION(), and tanh_deriv().

7.1.3.459 tanh_series()

```
ex GiNaC::tanh_series (
             const ex & x,
             const relational & rel,
             int order,
             unsigned options)  [static]
```

References cosh(), GINAC_ASSERT, I, is_a(), GiNaC::relational::lhs(), GiNaC::subs_options::no_pattern, GiNaC::info_flags::odd, Pi, sinh(), GiNaC::ex::subs(), and tanh_series().

Referenced by REGISTER_FUNCTION(), and tanh_series().

7.1.3.460 tanh_real_part()

```
ex GiNaC::tanh_real_part (
            const ex & x)  [static]
```

References imag_part(), GiNaC::power::power(), real_part(), tan(), tanh(), and tanh_real_part().

Referenced by REGISTER_FUNCTION(), and tanh_real_part().

7.1.3.461 tanh_imag_part()

```
ex GiNaC::tanh_imag_part (
            const ex & x)  [static]
```

References imag_part(), GiNaC::power::power(), real_part(), tan(), tanh(), and tanh_imag_part().

Referenced by REGISTER_FUNCTION(), and tanh_imag_part().

7.1.3.462 tanh_conjugate()

```
ex GiNaC::tanh_conjugate (
            const ex & x)  [static]
```

References GiNaC::ex::conjugate(), tanh(), and tanh_conjugate().

Referenced by REGISTER_FUNCTION(), and tanh_conjugate().

7.1.3.463 REGISTER_FUNCTION() [33/36]

```
GiNaC::REGISTER_FUNCTION (
            tanh ,
            eval_func(tanh_eval).  evalf_func(tanh_evalf).  info_func(atan_info).  derivative↩
_func(tanh_deriv).  series_func(tanh_series).  real_part_func(tanh_real_part).  imag_part_↩
func(tanh_imag_part).  conjugate_func(tanh_conjugate).  latex_name("\\tanh") )
```

References atan_info(), REGISTER_FUNCTION(), tanh(), tanh_conjugate(), tanh_deriv(), tanh_eval(), tanh_evalf(), tanh_imag_part(), tanh_real_part(), and tanh_series().

7.1.3.464 asinh_evalf()

```
ex GiNaC::asinh_evalf (
            const ex & x)  [static]
```

References asinh(), asinh_evalf(), ex_to(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by asinh_evalf(), and REGISTER_FUNCTION().

7.1.3.465 asinh_eval()

```
ex GiNaC::asinh_eval (
            const ex & x)  [static]
```

References _ex0, asinh(), asinh_eval(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::ex::is_zero(), GiNaC::info_flags::negative, and GiNaC::info_flags::numeric.

Referenced by asinh_eval(), and REGISTER_FUNCTION().

7.1.3.466 asinh_deriv()

```
ex GiNaC::asinh_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex1, _ex2, _ex_1_2, asinh_deriv(), GINAC_ASSERT, and GiNaC::power::power().

Referenced by asinh_deriv(), and REGISTER_FUNCTION().

7.1.3.467 asinh_conjugate()

```
ex GiNaC::asinh_conjugate (
            const ex & x)  [static]
```

References _num1_p, _num_1_p, asinh(), asinh_conjugate(), GiNaC::ex::conjugate(), ex_to(), GiNaC::ex::imag_part(), GiNaC::ex::info(), is_exactly_a(), GiNaC::numeric::is_zero(), GiNaC::info_flags::real, and GiNaC::ex::real_part().

Referenced by asinh_conjugate(), and REGISTER_FUNCTION().

7.1.3.468 REGISTER_FUNCTION() [34/36]

```
GiNaC::REGISTER_FUNCTION (
            asinh ,
            eval_func(asinh_eval).  evalf_func(asinh_evalf).  info_func(atan_info).  derivative↩
_func(asinh_deriv).  conjugate_func(asinh_conjugate) )
```

References asinh(), asinh_conjugate(), asinh_deriv(), asinh_eval(), asinh_evalf(), atan_info(), and REGISTER_FUNCTION().

7.1.3.469 acosh_evalf()

```
ex GiNaC::acosh_evalf (
            const ex & x)  [static]
```

References acosh(), acosh_evalf(), ex_to(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by acosh_evalf(), and REGISTER_FUNCTION().

7.1.3.470 acosh_eval()

```
ex GiNaC::acosh_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex_1, acosh(), acosh_eval(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), I, GiNaC::ex::info(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), and Pi.

Referenced by acosh_eval(), and REGISTER_FUNCTION().

7.1.3.471 acosh_deriv()

```
ex GiNaC::acosh_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex1, _ex_1, _ex_1_2, acosh_deriv(), GINAC_ASSERT, and GiNaC::power::power().

Referenced by acosh_deriv(), and REGISTER_FUNCTION().

7.1.3.472 acosh_conjugate()

```
ex GiNaC::acosh_conjugate (
            const ex & x)  [static]
```

References _num1_p, acosh(), acosh_conjugate(), GiNaC::ex::conjugate(), GiNaC::ex::imag_part(), is_exactly_a(), and GiNaC::ex::is_zero().

Referenced by acosh_conjugate(), and REGISTER_FUNCTION().

7.1.3.473 REGISTER_FUNCTION() [35/36]

```
GiNaC::REGISTER_FUNCTION (
            acosh ,
            eval_func(acosh_eval).  evalf_func(acosh_evalf).  info_func(asin_info).  derivative↩
_func(acosh_deriv).  conjugate_func(acosh_conjugate) )
```

References acosh(), acosh_conjugate(), acosh_deriv(), acosh_eval(), acosh_evalf(), asin_info(), and REGISTER_FUNCTION().

7.1.3.474 atanh_evalf()

```
ex GiNaC::atanh_evalf (
            const ex & x)  [static]
```

References atanh(), atanh_evalf(), ex_to(), GiNaC::basic::hold(), and is_exactly_a().

Referenced by atanh_evalf(), and REGISTER_FUNCTION().

7.1.3.475 atanh_eval()

```
ex GiNaC::atanh_eval (
            const ex & x)  [static]
```

References _ex0, _ex1, _ex_1, atanh(), atanh_eval(), GiNaC::info_flags::crational, ex_to(), GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, and GiNaC::pole_error::pole_error().

Referenced by atanh_eval(), and REGISTER_FUNCTION().

7.1.3.476 atanh_deriv()

```
ex GiNaC::atanh_deriv (
            const ex & x,
            unsigned deriv_param)  [static]
```

References _ex1, _ex2, _ex_1, atanh_deriv(), GINAC_ASSERT, and GiNaC::power::power().

Referenced by atanh_deriv(), and REGISTER_FUNCTION().

7.1.3.477 atanh_series()

```
ex GiNaC::atanh_series (
            const ex & arg,
            const relational & rel,
            int order,
            unsigned options)  [static]
```

References _ex0, _ex1, _ex1_2, _ex_1, _ex_1_2, abs(), atanh(), atanh_series(), csgn(), ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, I, is_a(), GiNaC::ex::is_equal(), GiNaC::relational::lhs(), log(), GiNaC::subs_options::no_pattern, GiNaC::ex::op(), Pi, GiNaC::pseries::pseries(), GiNaC::info_flags::real, GiNaC::relational::rhs(), series(), GiNaC::ex::subs(), and GiNaC::series_options::suppress_branchcut.

Referenced by atanh_series(), and REGISTER_FUNCTION().

7.1.3.478 atanh_conjugate()

```
ex GiNaC::atanh_conjugate (
            const ex & x)  [static]
```

References _num1_p, _num_1_p, atanh(), atanh_conjugate(), GiNaC::ex::conjugate(), GiNaC::ex::imag_part(), is_exactly_a(), and GiNaC::ex::is_zero().

Referenced by atanh_conjugate(), and REGISTER_FUNCTION().

7.1.3.479 REGISTER_FUNCTION() [36/36]

```
GiNaC::REGISTER_FUNCTION (
            atanh ,
            eval_func(atanh_eval).  evalf_func(atanh_evalf).  info_func(asin_info).  derivative↩
_func(atanh_deriv).  series_func(atanh_series).  conjugate_func(atanh_conjugate) )
```

References asin_info(), atanh(), atanh_conjugate(), atanh_deriv(), atanh_eval(), atanh_evalf(), atanh_series(), and REGISTER_FUNCTION().

7.1.3.480 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [12/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            integral ,
            basic ,
            print_func< print_dflt > &::do_print.  print_func< print_python > &::do_print.
print_func< print_latex > &::do_print_latex )
```

References GiNaC::integral::do_print(), GiNaC::integral::do_print_latex(), dynallocate(), and GINAC_IMPLEMENT_REGISTERED_CL

7.1.3.481 subsvalue()

```
ex GiNaC::subsvalue (
            const ex & var,
            const ex & value,
            const ex & fun)
```

References GiNaC::ex::evalf(), is_a(), GiNaC::ex::subs(), and subsvalue().

Referenced by adaptivesimpson(), and subsvalue().

7.1.3.482 adaptivesimpson()

```
GiNaC::ex GiNaC::adaptivesimpson (
            const ex & x,
            const ex & a_in,
            const ex & b_in,
            const ex & f,
            const ex & error)
```

Numeric integration routine based upon the "Adaptive Quadrature" one in "Numerical Analysis" by Burden and Faires.

Parameters are integration variable, left boundary, right boundary, function to be integrated and the relative integration error. The function should evalf into a number after substituting the integration variable by a number. Another thing to note is that this implementation is no good at integrating functions with discontinuities.

References abs(), adaptivesimpson(), GiNaC::error_and_integral::error_and_integral(), GiNaC::ex::evalf(), ex_to(), GiNaC::integral::integral(), is_exactly_a(), GiNaC::integral::max_integration_level, GiNaC::ex::subs(), and subsvalue().

Referenced by adaptivesimpson(), GiNaC::integral::evalf(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.483 GINAC_BIND_UNARCHIVER() [22/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            integral )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.484 GINAC_DECLARE_UNARCHIVER() [23/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            integral )
```

References adaptivesimpson(), GINAC_DECLARE_UNARCHIVER(), and GiNaC::integral::relative_integration_error.

7.1.3.485 ifactor()

```
ex GiNaC::ifactor (
            const numeric & n)
```

Returns the decomposition of the positive integer n into prime numbers in the form lst(lst(p1,...,pr), lst(a1,...,ar)) such that n = p1^a1 ... pr^ar.

References GiNaC::container< C >::append(), ifactor(), irem(), GiNaC::numeric::is_pos_integer(), and GiNaC::info_flags::prime.

Referenced by ifactor(), is_discriminant_of_quadratic_number_field(), and kronecker_symbol().

7.1.3.486 is_discriminant_of_quadratic_number_field()

```
bool GiNaC::is_discriminant_of_quadratic_number_field (
            const numeric & n)
```

Returns true if the integer n is either one or the discriminant of a quadratic number field.

Returns false otherwise.

Ref.: Toshitsune Miyake, Modular Forms, Chapter 3.1

References abs(), ex_to(), ifactor(), is_discriminant_of_quadratic_number_field(), GiNaC::numeric::is_odd(), mod(), GiNaC::container< C >::nops(), and GiNaC::container< C >::op().

Referenced by is_discriminant_of_quadratic_number_field().

7.1.3.487 kronecker_symbol()

```
numeric GiNaC::kronecker_symbol (
            const numeric & a,
            const numeric & n)
```

Returns the Kronecker symbol a: integer n: integer.

This routine defines kronecker_symbol(1,0) = 1 kronecker_symbol(-1,0) = 1 kronecker_symbol(a,0) = 0, a != 1,-1

In particular kronecker_symbol(-1,0) = 1 (in agreement with Sage)

Ref.: Toshitsune Miyake, Modular Forms, Chapter 3.1

References GiNaC::container< C >::begin(), GiNaC::container< C >::end(), ex_to(), ifactor(), GiNaC::numeric::is_even(), kronecker_symbol(), GiNaC::container< C >::op(), and pow().

Referenced by kronecker_symbol(), and primitive_dirichlet_character().

7.1.3.488 primitive_dirichlet_character()

```
numeric GiNaC::primitive_dirichlet_character (
            const numeric & n,
            const numeric & a)
```

Defines a primitive Dirichlet character through the Kronecker symbol.

n: integer a: discriminant of a quadratic field |a|: conductor

The character takes the values -1,0,1.

References kronecker_symbol(), and primitive_dirichlet_character().

Referenced by dirichlet_character(), generalised_Bernoulli_number(), and primitive_dirichlet_character().

7.1.3.489 dirichlet_character()

```
numeric GiNaC::dirichlet_character (
            const numeric & n,
            const numeric & a,
            const numeric & N)
```

Defines a Dirichlet character through the Kronecker symbol.

n: integer a: discriminant of a quadratic field |a|: conductor N: modulus, needs to be multiple of |a|

The character takes the values -1,0,1.

References dirichlet_character(), gcd(), and primitive_dirichlet_character().

Referenced by dirichlet_character().

7.1.3.490 generalised_Bernoulli_number()

```
numeric GiNaC::generalised_Bernoulli_number (
            const numeric & k,
            const numeric & b)
```

The generalised Bernoulli number.

k: index / modular weight

b: discriminant of a quadratic field, defines primitive character psi M=|b|: conductor of primitive character psi

The generalised Bernoulli number is computed from the series expansion of the generating function. The generating function is given in eq.(34), arXiv:1704.08895

References abs(), GiNaC::ex::coeff(), ex_to(), exp(), factorial(), generalised_Bernoulli_number(), primitive_dirichlet_character(), GiNaC::ex::series(), series_to_poly(), and GiNaC::numeric::to_int().

Referenced by generalised_Bernoulli_number().

7.1.3.491 Bernoulli_polynomial()

```
ex GiNaC::Bernoulli_polynomial (
             const numeric & k,
             const ex & x)
```

The Bernoulli polynomials.

References Bernoulli_polynomial(), GiNaC::ex::coeff(), exp(), factorial(), GiNaC::ex::series(), series_to_poly(), and GiNaC::numeric::to_int().

Referenced by Bernoulli_polynomial(), and GiNaC::Eisenstein_h_kernel::coefficient_a0().

7.1.3.492 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [13/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             integration_kernel ,
             basic ,
             print_func< print_context > &::do_print )
```

References GiNaC::integration_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.493 GINAC_BIND_UNARCHIVER() [23/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             integration_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.494 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [14/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             basic_log_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References GiNaC::basic_log_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.495 GINAC_BIND_UNARCHIVER() [24/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             basic_log_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.496 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [15/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             multiple_polylog_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References _ex1, GiNaC::multiple_polylog_kernel::do_print(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), and GiNaC::multiple_polylog_kernel::multiple_polylog_kernel().

7.1.3.497 GINAC_BIND_UNARCHIVER() [25/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             multiple_polylog_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.498 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [16/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             ELi_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References _ex0, GiNaC::ELi_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.499 GINAC_BIND_UNARCHIVER() [26/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             ELi_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.500 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [17/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             Ebar_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References _ex0, GiNaC::Ebar_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.501 GINAC_BIND_UNARCHIVER() [27/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             Ebar_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.502 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [18/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             Kronecker_dtau_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References _ex0, _ex1, GiNaC::Kronecker_dtau_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.503 GINAC_BIND_UNARCHIVER() [28/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             Kronecker_dtau_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.504 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [19/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             Kronecker_dz_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References _ex0, _ex1, GiNaC::Kronecker_dz_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.505 GINAC_BIND_UNARCHIVER() [29/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             Kronecker_dz_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.506 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [20/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             Eisenstein_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References _ex0, _ex1, GiNaC::Eisenstein_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.507 GINAC_BIND_UNARCHIVER() [30/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             Eisenstein_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.508 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [21/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             Eisenstein_h_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References _ex0, _ex1, GiNaC::Eisenstein_h_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.509 GINAC_BIND_UNARCHIVER() [31/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             Eisenstein_h_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.510 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [22/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             modular_form_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References _ex0, _ex1, GiNaC::modular_form_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.511 GINAC_BIND_UNARCHIVER() [32/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             modular_form_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.512 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [23/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             user_defined_kernel ,
             integration_kernel ,
             print_func< print_context > &::do_print )
```

References _ex0, GiNaC::user_defined_kernel::do_print(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.513 GINAC_BIND_UNARCHIVER() [33/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             user_defined_kernel )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.514 GINAC_DECLARE_UNARCHIVER() [24/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             integration_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.515 GINAC_DECLARE_UNARCHIVER() [25/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             basic_log_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.516 GINAC_DECLARE_UNARCHIVER() [26/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             multiple_polylog_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.517 GINAC_DECLARE_UNARCHIVER() [27/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             ELi_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.518 GINAC_DECLARE_UNARCHIVER() [28/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             Ebar_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.519 GINAC_DECLARE_UNARCHIVER() [29/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             Kronecker_dtau_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.520 GINAC_DECLARE_UNARCHIVER() [30/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             Kronecker_dz_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.521 GINAC_DECLARE_UNARCHIVER() [31/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            Eisenstein_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.522 GINAC_DECLARE_UNARCHIVER() [32/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            Eisenstein_h_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.523 GINAC_DECLARE_UNARCHIVER() [33/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            modular_form_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.524 GINAC_DECLARE_UNARCHIVER() [34/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            user_defined_kernel )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.525 GINAC_DECLARE_UNARCHIVER() [35/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            lst )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.526 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [24/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            matrix ,
            basic ,
            print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_tree > &::do_print_tree.  print_func< print_python_repr >
&::do_print_python_repr )
```

Default ctor.

Initializes to 1 x 1-dimensional zero-matrix.

References _ex0, GiNaC::matrix::do_print(), GiNaC::matrix::do_print_latex(), GiNaC::matrix::do_print_python_repr(), GiNaC::basic::do_print_tree(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), and GiNaC::status_flags::not_shareable.

7.1.3.527 GINAC_BIND_UNARCHIVER() [34/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             matrix )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.528 lst_to_matrix()

```
ex GiNaC::lst_to_matrix (
             const lst & l)
```

Convert list of lists to matrix.

References cols(), dynallocate(), ex_to(), is_a(), lst_to_matrix(), GiNaC::container< C >::nops(), and rows().

Referenced by lst_to_matrix().

7.1.3.529 diag_matrix() [1/2]

```
ex GiNaC::diag_matrix (
             const lst & l)
```

Convert list of diagonal elements to matrix.

References diag_matrix(), dynallocate(), and GiNaC::container< C >::nops().

Referenced by diag_matrix(), and diag_matrix().

7.1.3.530 diag_matrix() [2/2]

```
ex GiNaC::diag_matrix (
             std::initializer_list< ex > l)
```

References diag_matrix(), and dynallocate().

7.1.3.531 unit_matrix() [1/2]

```
ex GiNaC::unit_matrix (
             unsigned r,
             unsigned c)
```

Create an r times c unit matrix.

References _ex1, dynallocate(), GiNaC::status_flags::evaluated, GiNaC::basic::setflag(), and unit_matrix().

Referenced by unit_matrix(), and unit_matrix().

7.1.3.532 symbolic_matrix() [1/2]

```
ex GiNaC::symbolic_matrix (
             unsigned r,
             unsigned c,
             const std::string & base_name,
             const std::string & tex_base_name)
```

Create an r times c matrix of newly generated symbols consisting of the given base name plus the numeric row/↩ column position of each element.

The base name for LaTeX output is specified separately.

References dynallocate(), GiNaC::status_flags::evaluated, GiNaC::basic::setflag(), GiNaC::symbol::symbol(), and symbolic_matrix().

Referenced by symbolic_matrix(), and symbolic_matrix().

7.1.3.533 reduced_matrix()

```
ex GiNaC::reduced_matrix (
             const matrix & m,
             unsigned r,
             unsigned c)
```

Return the reduced matrix that is formed by deleting the rth row and cth column of matrix m.

The determinant of the result is the Minor r, c.

References cols(), GiNaC::matrix::cols(), dynallocate(), GiNaC::status_flags::evaluated, reduced_matrix(), GiNaC::matrix::rows(), rows(), and GiNaC::basic::setflag().

Referenced by reduced_matrix().

7.1.3.534 sub_matrix()

```
ex GiNaC::sub_matrix (
             const matrix & m,
             unsigned r,
             unsigned nr,
             unsigned c,
             unsigned nc)
```

Return the nr times nc submatrix starting at position r, c of matrix m.

References GiNaC::matrix::cols(), dynallocate(), GiNaC::status_flags::evaluated, GiNaC::matrix::rows(), GiNaC::basic::setflag(), and sub_matrix().

Referenced by lst_to_clifford(), and sub_matrix().

7.1.3.535 GINAC_DECLARE_UNARCHIVER() [36/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             matrix )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.536 nops() [2/2]

```
size_t GiNaC::nops (
            const matrix & m)  [inline]
```

References GiNaC::matrix::nops(), and nops().

7.1.3.537 expand() [2/2]

```
ex GiNaC::expand (
            const matrix & m,
            unsigned options = 0)  [inline]
```

References GiNaC::basic::expand(), and expand().

7.1.3.538 evalf() [2/2]

```
ex GiNaC::evalf (
            const matrix & m)  [inline]
```

References GiNaC::basic::evalf(), and evalf().

7.1.3.539 rows()

```
unsigned GiNaC::rows (
            const matrix & m)  [inline]
```

References GiNaC::matrix::rows(), and rows().

Referenced by clifford_moebius_map(), clifford_moebius_map(), lst_to_clifford(), lst_to_matrix(), reduced_matrix(), and rows().

7.1.3.540 cols()

```
unsigned GiNaC::cols (
            const matrix & m)  [inline]
```

References cols(), and GiNaC::matrix::cols().

Referenced by clifford_moebius_map(), clifford_moebius_map(), cols(), lst_to_clifford(), lst_to_matrix(), and reduced_matrix().

7.1.3.541 transpose()

```
matrix GiNaC::transpose (
            const matrix & m)  [inline]
```

References GiNaC::matrix::transpose(), and transpose().

Referenced by transpose().

7.1.3.542 determinant()

```
ex GiNaC::determinant (
             const matrix & m,
             unsigned options = determinant_algo::automatic)  [inline]
```

References GiNaC::determinant_algo::automatic, determinant(), and GiNaC::matrix::determinant().

Referenced by determinant().

7.1.3.543 trace()

```
ex GiNaC::trace (
             const matrix & m)  [inline]
```

References GiNaC::matrix::trace(), and trace().

Referenced by trace().

7.1.3.544 charpoly()

```
ex GiNaC::charpoly (
             const matrix & m,
             const ex & lambda)  [inline]
```

References charpoly(), and GiNaC::matrix::charpoly().

Referenced by charpoly().

7.1.3.545 inverse() [1/3]

```
matrix GiNaC::inverse (
             const matrix & m)  [inline]
```

References GiNaC::solve_algo::automatic, inverse(), and GiNaC::matrix::inverse().

Referenced by inverse(), inverse(), and inverse().

7.1.3.546 inverse() [2/3]

```
matrix GiNaC::inverse (
             const matrix & m,
             unsigned algo)  [inline]
```

References inverse(), and GiNaC::matrix::inverse().

7.1.3.547 rank() [1/2]

```
unsigned GiNaC::rank (
            const matrix & m)  [inline]
```

References GiNaC::matrix::rank(), and rank().

Referenced by rank(), and rank().

7.1.3.548 rank() [2/2]

```
unsigned GiNaC::rank (
            const matrix & m,
            unsigned solve_algo)  [inline]
```

References GiNaC::matrix::rank(), and rank().

7.1.3.549 unit_matrix() [2/2]

```
ex GiNaC::unit_matrix (
            unsigned x)  [inline]
```

Create a x times x unit matrix.

References unit_matrix().

7.1.3.550 symbolic_matrix() [2/2]

```
ex GiNaC::symbolic_matrix (
            unsigned r,
            unsigned c,
            const std::string & base_name)  [inline]
```

Create an r times c matrix of newly generated symbols consisting of the given base name plus the numeric row/↩ column position of each element.

References symbolic_matrix().

7.1.3.551 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [25/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            mul ,
            expairseq ,
            print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_csrc > &::do_print_csrc.  print_func< print_tree > &::do_↩
print_tree.  print_func< print_python_repr > &::do_print_python_repr )
```

References GiNaC::mul::do_print(), GiNaC::mul::do_print_csrc(), GiNaC::mul::do_print_latex(), GiNaC::mul::do_print_python_repr(), GiNaC::basic::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.552 tryfactsubs()

```
bool GiNaC::tryfactsubs (
             const ex & origfactor,
             const ex & patternfactor,
             int & nummatches,
             exmap & repls)
```

References ex_to(), GiNaC::ex::info(), GiNaC::info_flags::integer, is_exactly_a(), GiNaC::ex::match(), GiNaC::ex::op(), and tryfactsubs().

Referenced by algebraic_match_mul_with_mul(), GiNaC::mul::algebraic_subs_mul(), GiNaC::power::subs(), and tryfactsubs().

7.1.3.553 algebraic_match_mul_with_mul()

```
bool GiNaC::algebraic_match_mul_with_mul (
             const mul & e,
             const ex & pat,
             exmap & repls,
             int factor,
             int & nummatches,
             const std::vector< bool > & subsed,
             std::vector< bool > & matched)
```

Checks whether e matches to the pattern pat and the (possibly to be updated) list of replacements repls.

This matching is in the sense of algebraic substitutions. Matching starts with pat.op(factor) of the pattern because the factors before this one have already been matched. The (possibly updated) number of matches is in nummatches. subsed[i] is true for factors that already have been replaced by previous substitutions and matched[i] is true for factors that have been matched by the current match.

References algebraic_match_mul_with_mul(), factor(), GINAC_ASSERT, GiNaC::ex::nops(), GiNaC::expairseq::nops(), GiNaC::ex::op(), GiNaC::expairseq::op(), and tryfactsubs().

Referenced by algebraic_match_mul_with_mul(), GiNaC::mul::algebraic_subs_mul(), and GiNaC::mul::has().

7.1.3.554 GINAC_BIND_UNARCHIVER() [35/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             mul )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.555 GINAC_DECLARE_UNARCHIVER() [37/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
             mul )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.556 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [26/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            ncmul ,
            exprseq ,
            print_func< print_context > &::do_print.  print_func< print_tree > &::do_print↩
_tree.  print_func< print_csrc > &::do_print_csrc.  print_func< print_python_repr > &::do_↩
print_csrc )
```

References GiNaC::ncmul::do_print(), GiNaC::ncmul::do_print_csrc(), GiNaC::basic::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.557 reeval_ncmul()

```
ex GiNaC::reeval_ncmul (
            const exvector & v)
```

Referenced by GiNaC::clifford::eval_ncmul(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.558 hold_ncmul()

```
ex GiNaC::hold_ncmul (
            const exvector & v)
```

Referenced by GiNaC::basic::eval_ncmul(), GiNaC::clifford::eval_ncmul(), GiNaC::color::eval_ncmul(), GiNaC::structure< T, Comparis and GINAC_DECLARE_UNARCHIVER().

7.1.3.559 GINAC_BIND_UNARCHIVER() [36/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            ncmul )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.560 GINAC_DECLARE_UNARCHIVER() [38/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            ncmul )
```

References GINAC_DECLARE_UNARCHIVER(), hold_ncmul(), and reeval_ncmul().

7.1.3.561 get_first_symbol()

```
bool GiNaC::get_first_symbol (
             const ex & e,
             ex & x)  [static]
```

Return pointer to first symbol found in expression.

Due to GiNaC's internal ordering of terms, it may not be obvious which symbol this function returns for a given expression.

Parameters

e	expression to search
x	first symbol found (returned)

Returns

"false" if no symbol was found, "true" otherwise

References get_first_symbol(), is_a(), is_exactly_a(), GiNaC::ex::nops(), and GiNaC::ex::op().

Referenced by divide(), frac_cancel(), get_first_symbol(), and GiNaC::ex::unit().

7.1.3.562 add_symbol()

```
void GiNaC::add_symbol (
             const ex & s,
             sym_desc_vec & v)  [static]
```

References add_symbol(), and GiNaC::sym_desc::sym_desc().

Referenced by add_symbol(), and collect_symbols().

7.1.3.563 collect_symbols()

```
void GiNaC::collect_symbols (
             const ex & e,
             sym_desc_vec & v)  [static]
```

References add_symbol(), collect_symbols(), is_a(), is_exactly_a(), GiNaC::ex::nops(), and GiNaC::ex::op().

Referenced by collect_symbols(), and get_symbol_stats().

7.1.3.564 get_symbol_stats()

```
void GiNaC::get_symbol_stats (
            const ex & a,
            const ex & b,
            sym_desc_vec & v)  [static]
```

Collect statistical information about symbols in polynomials.

This function fills in a vector of "sym_desc" structs which contain information about the highest and lowest degrees of all symbols that appear in two polynomials. The vector is then sorted by minimum degree (lowest to highest). The information gathered by this function is used by the GCD routines to identify trivial factors and to determine which variable to choose as the main variable for GCD computation.

Parameters

a	first multivariate polynomial
b	second multivariate polynomial
v	vector of sym_desc structs (filled in)

References collect_symbols(), GiNaC::ex::degree(), get_symbol_stats(), GiNaC::ex::lcoeff(), GiNaC::ex::ldegree(), and GiNaC::ex::nops().

Referenced by divide_in_z(), gcd(), get_symbol_stats(), and sqrfree().

7.1.3.565 lcmcoeff()

```
numeric GiNaC::lcmcoeff (
            const ex & e,
            const numeric & l)  [static]
```

References _num1_p, denom(), ex_to(), GiNaC::ex::info(), is_a(), is_exactly_a(), lcm(), lcmcoeff(), GiNaC::ex::nops(), GiNaC::ex::op(), pow(), and GiNaC::info_flags::rational.

Referenced by heur_gcd(), lcm_of_coefficients_denominators(), lcmcoeff(), and multiply_lcm().

7.1.3.566 lcm_of_coefficients_denominators()

```
numeric GiNaC::lcm_of_coefficients_denominators (
            const ex & e)  [static]
```

Compute LCM of denominators of coefficients of a polynomial.

Given a polynomial with rational coefficients, this function computes the LCM of the denominators of all coefficients. This can be used to bring a polynomial from Q[X] to Z[X].

Parameters

e	multivariate polynomial (need not be expanded)

Returns

LCM of denominators of coefficients

References _num1_p, lcm_of_coefficients_denominators(), and lcmcoeff().

Referenced by frac_cancel(), heur_gcd(), lcm_of_coefficients_denominators(), and sqrfree().

7.1.3.567 multiply_lcm()

```
ex GiNaC::multiply_lcm (
             const ex & e,
             const numeric & lcm)  [static]
```

Bring polynomial from Q[X] to Z[X] by multiplying in the previously determined LCM of the coefficient's denominators.

Parameters

e	multivariate polynomial (need not be expanded)
lcm	LCM to multiply in

References _num1_p, dynallocate(), ex_to(), is_a(), GiNaC::ex::is_equal(), is_exactly_a(), GiNaC::numeric::is_rational(), lcm(), lcmcoeff(), multiply_lcm(), GiNaC::ex::nops(), GiNaC::ex::op(), and pow().

Referenced by frac_cancel(), multiply_lcm(), and sqrfree().

7.1.3.568 quo()

```
ex GiNaC::quo (
             const ex & a,
             const ex & b,
             const ex & x,
             bool check_args)
```

Quotient q(x) of polynomials a(x) and b(x) in Q[x].

It satisfies a(x)=b(x)∗q(x)+r(x).

Parameters

a	first polynomial in x (dividend)
b	second polynomial in x (divisor)
x	a and b are polynomials in x
check_args	check whether a and b are polynomials with rational coefficients (defaults to "true")

Returns

quotient of a and b in Q[x]

References _ex1, GiNaC::ex::coeff(), GiNaC::ex::degree(), divide(), dynallocate(), GiNaC::ex::expand(), expand(), GiNaC::ex::info(), GiNaC::ex::is_equal(), is_exactly_a(), GiNaC::ex::is_zero(), pow(), quo(), and GiNaC::info_flags::rational_polynomial.

Referenced by decomp_rational(), GiNaC::ex::primpart(), quo(), sqrfree_parfrac(), sqrfree_yun(), and GiNaC::ex::unitcontprim().

7.1.3.569 rem()

```
ex GiNaC::rem (
              const ex & a,
              const ex & b,
              const ex & x,
              bool check_args)
```

Remainder r(x) of polynomials a(x) and b(x) in Q[x].

It satisfies a(x)=b(x)∗q(x)+r(x).

Parameters

a	first polynomial in x (dividend)
b	second polynomial in x (divisor)
x	a and b are polynomials in x
check_args	check whether a and b are polynomials with rational coefficients (defaults to "true")

Returns

remainder of a(x) and b(x) in Q[x]

References _ex0, GiNaC::ex::coeff(), GiNaC::ex::degree(), divide(), dynallocate(), GiNaC::ex::expand(), expand(), GiNaC::ex::info(), GiNaC::ex::is_equal(), is_exactly_a(), GiNaC::ex::is_zero(), pow(), GiNaC::info_flags::rational_polynomial, and rem().

Referenced by decomp_rational(), rem(), and sqrfree_parfrac().

7.1.3.570 decomp_rational()

```
ex GiNaC::decomp_rational (
              const ex & a,
              const ex & x)
```

Decompose rational function a(x)=N(x)/D(x) into P(x)+n(x)/D(x) with degree(n, x) $<$ degree(D, x).

Parameters

a	rational function in x
x	a is a function of x

Returns

decomposed function.

References decomp_rational(), denom(), is_exactly_a(), numer(), numer_denom(), GiNaC::ex::op(), quo(), and rem().

Referenced by decomp_rational().

7.1.3.571 prem()

```
ex GiNaC::prem (
            const ex & a,
            const ex & b,
            const ex & x,
            bool check_args)
```

Pseudo-remainder of polynomials a(x) and b(x) in Q[x].

Parameters

a	first polynomial in x (dividend)
b	second polynomial in x (divisor)
x	a and b are polynomials in x
check_args	check whether a and b are polynomials with rational coefficients (defaults to "true")

Returns

pseudo-remainder of a(x) and b(x) in Q[x]

References _ex0, _ex1, GiNaC::ex::coeff(), GiNaC::ex::degree(), GiNaC::ex::expand(), expand(), GiNaC::ex::info(), is_exactly_a(), GiNaC::ex::is_zero(), pow(), prem(), and GiNaC::info_flags::rational_polynomial.

Referenced by prem(), and sr_gcd().

7.1.3.572 sprem()

```
ex GiNaC::sprem (
            const ex & a,
            const ex & b,
            const ex & x,
            bool check_args)
```

Sparse pseudo-remainder of polynomials a(x) and b(x) in Q[x].

Parameters

a	first polynomial in x (dividend)
b	second polynomial in x (divisor)
x	a and b are polynomials in x
check_args	check whether a and b are polynomials with rational coefficients (defaults to "true")

Returns

sparse pseudo-remainder of a(x) and b(x) in Q[x]

References _ex0, _ex1, GiNaC::ex::coeff(), GiNaC::ex::degree(), GiNaC::ex::expand(), expand(), GiNaC::ex::info(), is_exactly_a(), GiNaC::ex::is_zero(), pow(), GiNaC::info_flags::rational_polynomial, and sprem().

Referenced by sprem().

7.1.3.573 divide()

```
bool GiNaC::divide (
            const ex & a,
            const ex & b,
            ex & q,
            bool check_args)
```

Exact polynomial division of a(X) by b(X) in Q[X].

Parameters

a	first multivariate polynomial (dividend)
b	second multivariate polynomial (divisor)
q	quotient (returned)
check_args	check whether a and b are polynomials with rational coefficients (defaults to "true")

Returns

"true" when exact division succeeds (quotient returned in q), "false" otherwise (q left untouched)

References _ex0, _ex1, GiNaC::ex::coeff(), GiNaC::ex::degree(), divide(), dynallocate(), ex_to(), GiNaC::ex::expand(), expand(), get_first_symbol(), GiNaC::ex::info(), GiNaC::ex::is_equal(), is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::ex::nops(), GiNaC::ex::op(), pow(), and GiNaC::info_flags::rational_polynomial.

Referenced by divide(), find_common_factor(), GiNaC::matrix::fraction_free_elimination(), gcd(), quo(), and rem().

7.1.3.574 divide_in_z()

```
bool GiNaC::divide_in_z (
            const ex & a,
            const ex & b,
            ex & q,
            sym_desc_vec::const_iterator var)  [static]
```

Exact polynomial division of a(X) by b(X) in Z[X].

This functions works like divide() but the input and output polynomials are in Z[X] instead of Q[X] (i.e. they have integer coefficients). Unlike divide(), it doesn't check whether the input polynomials really are integer polynomials, so be careful of what you pass in. Also, you have to run get_symbol_stats() over the input polynomials before calling this function and pass an iterator to the first element of the sym_desc vector. This function is used internally by the heur_gcd().

Parameters

a	first multivariate polynomial (dividend)
b	second multivariate polynomial (divisor)
q	quotient (returned)
var	iterator to first element of vector of sym_desc structs

Returns

"true" when exact division succeeds (the quotient is returned in q), "false" otherwise.

See also

get_symbol_stats, heur_gcd

References _ex0, _ex1, _num0_p, _num1_p, GiNaC::ex::coeff(), GiNaC::ex::degree(), divide_in_z(), dynallocate(), ex_to(), GiNaC::ex::expand(), expand(), GiNaC::ex::find(), get_symbol_stats(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::numeric::inverse(), GiNaC::ex::is_equal(), is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::subs_options::no_pattern, GiNaC::ex::op(), pow(), and GiNaC::ex::subs().

Referenced by divide_in_z(), heur_gcd_z(), and sr_gcd().

7.1.3.575 sr_gcd()

```
ex GiNaC::sr_gcd (
            const ex & a,
            const ex & b,
            sym_desc_vec::const_iterator var)  [static]
```

Compute GCD of multivariate polynomials using the subresultant PRS algorithm.

This function is used internally by gcd().

Parameters

a	first multivariate polynomial
b	second multivariate polynomial
var	iterator to first element of vector of sym_desc structs

Returns

the GCD as a new expression

See also

gcd

References _ex0, _ex1, GiNaC::ex::content(), GiNaC::ex::degree(), divide_in_z(), GiNaC::ex::expand(), gcd(), is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::ex::lcoeff(), pow(), prem(), GiNaC::ex::primpart(), psi(), and sr_gcd().

Referenced by gcd(), and sr_gcd().

7.1.3.576 interpolate()

```
ex GiNaC::interpolate (
            const ex & gamma,
            const numeric & xi,
            const ex & x,
            int degree_hint = 1)  [static]
```

xi-adic polynomial interpolation

References dynallocate(), interpolate(), GiNaC::numeric::inverse(), GiNaC::ex::is_zero(), pow(), and GiNaC::ex::smod().

Referenced by heur_gcd_z(), and interpolate().

7.1.3.577 heur_gcd_z()

```
bool GiNaC::heur_gcd_z (
            ex & res,
            const ex & a,
            const ex & b,
            ex * ca,
            ex * cb,
            sym_desc_vec::const_iterator var)  [static]
```

Compute GCD of multivariate polynomials using the heuristic GCD algorithm.

get_symbol_stats() must have been called previously with the input polynomials and an iterator to the first element of the sym_desc vector passed in. This function is used internally by gcd().

Parameters

a	first integer multivariate polynomial (expanded)
b	second integer multivariate polynomial (expanded)
ca	cofactor of polynomial a (returned), nullptr to suppress calculation of cofactor
cb	cofactor of polynomial b (returned), nullptr to suppress calculation of cofactor
var	iterator to first element of vector of sym_desc structs
res	the GCD (returned)

Returns

true if GCD was computed, false otherwise.

See also

gcd

Exceptions

gcdheu_failed()	

References GiNaC::ex::degree(), divide_in_z(), ex_to(), GiNaC::ex::expand(), gcd(), heur_gcd_z(), GiNaC::numeric::int_length(), GiNaC::ex::integer_content(), interpolate(), GiNaC::numeric::inverse(), iquo(), is_exactly_a(), GiNaC::ex::is_zero(), isqrt(), GiNaC::ex::max_coefficient(), GiNaC::subs_options::no_pattern, GiNaC::numeric::numeric(), and GiNaC::ex::subs().

Referenced by heur_gcd(), and heur_gcd_z().

7.1.3.578 heur_gcd()

```
bool GiNaC::heur_gcd (
              ex & res,
              const ex & a,
              const ex & b,
              ex * ca,
              ex * cb,
              sym_desc_vec::const_iterator var)  [static]
```

Compute GCD of multivariate polynomials using the heuristic GCD algorithm.

get_symbol_stats() must have been called previously with the input polynomials and an iterator to the first element of the sym_desc vector passed in. This function is used internally by gcd().

Parameters

a	first rational multivariate polynomial (expanded)
b	second rational multivariate polynomial (expanded)
ca	cofactor of polynomial a (returned), nullptr to suppress calculation of cofactor
cb	cofactor of polynomial b (returned), nullptr to suppress calculation of cofactor
var	iterator to first element of vector of sym_desc structs
res	the GCD (returned)

Returns

true if GCD was computed, false otherwise.

See also

heur_gcd_z

gcd

References heur_gcd(), heur_gcd_z(), GiNaC::ex::info(), GiNaC::info_flags::integer_polynomial, lcm_of_coefficients_denominators(), and lcmcoeff().

Referenced by gcd(), and heur_gcd().

7.1.3.579 gcd_pf_pow()

```
ex GiNaC::gcd_pf_pow (
              const ex & a,
              const ex & b,
              ex * ca,
              ex * cb)  [static]
```

References _ex1, ex_to(), expand(), gcd(), gcd_pf_pow(), gcd_pf_pow_pow(), GINAC_ASSERT, is_a(), GiNaC::ex::is_equal(), is_exactly_a(), GiNaC::ex::ldegree(), GiNaC::ex::op(), and pow().

Referenced by gcd(), and gcd_pf_pow().

7.1.3.580 gcd_pf_mul()

```
ex GiNaC::gcd_pf_mul (
              const ex & a,
              const ex & b,
              ex * ca,
              ex * cb)  [static]
```

References dynallocate(), gcd(), gcd_pf_mul(), GINAC_ASSERT, is_exactly_a(), GiNaC::ex::nops(), and GiNaC::ex::op().

Referenced by gcd(), and gcd_pf_mul().

7.1.3.581 gcd() [1/2]

```
ex GiNaC::gcd (
              const ex & a,
              const ex & b,
              ex * ca,
              ex * cb,
              bool check_args,
              unsigned options)
```

Compute GCD (Greatest Common Divisor) of multivariate polynomials a(X) and b(X) in Z[X].

Optionally also compute the cofactors of a and b, defined by a = ca ∗ gcd(a, b) and b = cb ∗ gcd(a, b).

Parameters

a	first multivariate polynomial
b	second multivariate polynomial
ca	pointer to expression that will receive the cofactor of a, or nullptr
cb	pointer to expression that will receive the cofactor of b, or nullptr
check_args	check whether a and b are polynomials with rational coefficients (defaults to "true")
options	see GiNaC::gcd_options

Returns

the GCD as a new expression

References _ex0, _ex1, divide(), ex_to(), GiNaC::ex::expand(), expand(), gcd(), gcd_pf_mul(), gcd_pf_pow(), get_symbol_stats(), heur_gcd(), GiNaC::ex::info(), GiNaC::ex::integer_content(), is_a(), GiNaC::ex::is_equal(), is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::numeric::is_zero(), GiNaC::gcd_options::no_heur_gcd, GiNaC::gcd_options::no_part_fa GiNaC::subs_options::no_pattern, pow(), GiNaC::info_flags::rational_polynomial, sr_gcd(), GiNaC::ex::subs(), GiNaC::ex::unitcontprim(), and GiNaC::gcd_options::use_sr_gcd.

Referenced by GiNaC::ex::content(), dirichlet_character(), find_common_factor(), frac_cancel(), gcd(), gcd(), gcd_pf_mul(), gcd_pf_pow(), gcd_pf_pow_pow(), GINAC_DECLARE_UNARCHIVER(), heur_gcd_z(), GiNaC::add::integer_content(), lcm(), GiNaC::add::normal(), sqrfree_yun(), and sr_gcd().

7.1.3.582 gcd_pf_pow_pow()

```
ex GiNaC::gcd_pf_pow_pow (
             const ex & a,
             const ex & b,
             ex * ca,
             ex * cb)  [static]
```

References _ex1, gcd(), gcd_pf_pow_pow(), GiNaC::ex::is_equal(), GiNaC::ex::op(), and pow().

Referenced by gcd_pf_pow(), and gcd_pf_pow_pow().

7.1.3.583 lcm() [1/2]

```
ex GiNaC::lcm (
             const ex & a,
             const ex & b,
             bool check_args)
```

Compute LCM (Least Common Multiple) of multivariate polynomials in Z[X].

Parameters

a	first multivariate polynomial
b	second multivariate polynomial
check_args	check whether a and b are polynomials with rational coefficients (defaults to "true")

Returns

the LCM as a new expression

References ex_to(), gcd(), GiNaC::ex::info(), is_exactly_a(), lcm(), and GiNaC::info_flags::rational_polynomial.

Referenced by GINAC_DECLARE_UNARCHIVER(), GiNaC::add::integer_content(), lcm(), lcm(), lcmcoeff(), multiply_lcm(), and sqrfree().

7.1.3.584 sqrfree_yun()

```
epvector GiNaC::sqrfree_yun (
             const ex & a,
             const symbol & x)  [static]
```

Compute square-free factorization of multivariate polynomial a(x) using Yun's algorithm.

Used internally by sqrfree().

Parameters

a	multivariate polynomial over Z[X], treated here as univariate polynomial in x (needs not be expanded).

x	variable to factor in

Returns

vector of expairs (factor, exponent), sorted by exponents

References _ex1, GiNaC::ex::diff(), GiNaC::expair::expair(), gcd(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), quo(), and sqrfree_yun().

Referenced by sqrfree(), sqrfree_parfrac(), and sqrfree_yun().

7.1.3.585 sqrfree()

```
ex GiNaC::sqrfree (
            const ex & a,
            const lst & l)
```

Compute a square-free factorization of a multivariate polynomial in Q[X].

Parameters

a	multivariate polynomial over Q[X] (needs not be expanded)
l	lst of variables to factor in, may be left empty for autodetection

Returns

a square-free factorization of a.

Note

A polynomial $p(X) \in C[X]$ is said *square-free* if, whenever any two polynomials $q(X)$ and $r(X)$ are such that

$$p(X) = q(X)^2 r(X),$$

we have $q(X) \in C$. This means that $p(X)$ has no repeated factors, apart eventually from constants. Given a polynomial $p(X) \in C[X]$, we say that the decomposition

$$p(X) = b \cdot p_1(X)^{a_1} \cdot p_2(X)^{a_2} \cdots p_r(X)^{a_r}$$

is a *square-free factorization* of $p(X)$ if the following conditions hold:

1. $b \in C$ and $b \neq 0$;
2. a_i is a positive integer for $i = 1, \ldots, r$;
3. the degree of the polynomial p_i is strictly positive for $i = 1, \ldots, r$;
4. the polynomial $\Pi_{i=1}^{r} p_i(X)$ is square-free.

Square-free factorizations need not be unique. For example, if a_i is even, we could change the polynomial $p_i(X)$ into $-p_i(X)$. Observe also that the factors $p_i(X)$ need not be irreducible polynomials.

References _ex0, GiNaC::container< C >::append(), ex_to(), get_symbol_stats(), is_a(), is_exactly_a(), lcm(), lcm_of_coefficients_denominators(), GiNaC::mul::mul(), multiply_lcm(), GiNaC::container< C >::nops(), GiNaC::container< C >::op(), GiNaC::container< C >::remove_first(), sqrfree(), and sqrfree_yun().

Referenced by sqrfree().

7.1.3.586 sqrfree_parfrac()

```
ex GiNaC::sqrfree_parfrac (
            const ex & a,
            const symbol & x)
```

Compute square-free partial fraction decomposition of rational function a(x).

Parameters

a	rational function over Z[x], treated as univariate polynomial in x
x	variable to factor in

Returns

decomposed rational function

References _ex1, GiNaC::basic::coeff(), coeff(), GiNaC::ex::coeff(), degree(), denom(), ex_to(), GiNaC::ex::expand(), factor(), GINAC_ASSERT, numer(), numer_denom(), GiNaC::ex::op(), pow(), quo(), rem(), rhs(), GiNaC::matrix::solve(), sqrfree_parfrac(), sqrfree_yun(), GiNaC::symbol::symbol(), and to_int().

Referenced by sqrfree_parfrac().

7.1.3.587 replace_with_symbol() [1/2]

```
ex GiNaC::replace_with_symbol (
            const ex & e,
            exmap & repl,
            exmap & rev_lookup,
            lst & modifier)  [static]
```

Create a symbol for replacing the expression "e" (or return a previously assigned symbol).

The symbol and expression are appended to repl, for a later application of subs(). An entry in the replacement table repl can be changed in some cases. If it was altered, we need to provide the modifier for the previously build expressions. The modifier is an (ordered) list, because those substitutions need to be done in the incremental order. As an example let us consider a rationalisation of the expression e = exp(2∗x)∗cos(exp(2∗x)+1)∗exp(x) The first factor GiNaC denotes by something like symbol1 and will record: e =symbol1∗cos(symbol1 + 1)∗exp(x) repl = {symbol1 : exp(2∗x)} Similarly, the second factor would be denoted as symbol2 and we will have e =symbol1∗symbol2∗exp(x) repl = {symbol1 : exp(2∗x), symbol2 : cos(symbol1 + 1)} Denoting the third term as symbol3 GiNaC is willing to re-think exp(2∗x) as symbol3^2 rather than just symbol1. Here are two issues: 1) The replacement "symbol1 -> symbol3^2" in the previous part of the expression needs to be done outside of the present routine; 2) The pair "symbol1 : exp(2∗x)" shall be deleted from the replacement table repl. However, this will create illegal substitution "symbol2 : cos(symbol1 + 1)" with undefined symbol1. These both problems are mitigated through the additions of the record "symbol1==symbol3^2" to the list modifier. Changed length of the modifier signals to the calling code that the previous portion of the expression needs to be altered (it solves 1). Thus GiNaC can record now e =symbol3^2∗symbol2∗symbol3 repl = {symbol2 : cos(symbol1 + 1), symbol3 : exp(x)} modifier = {symbol1==symbol3^2} Then, doing the backward substitutions the list modifier will be used to restore such iterative substitutions in the right way (this solves 2).

See also

ex::normal

References _ex_1, GiNaC::container< C >::append(), degree(), denom(), dynallocate(), ex_to(), exp(), GiNaC::ex::find(), is_a(), GiNaC::ex::is_equal(), is_ex_the_function, is_integer(), is_rational(), GiNaC::subs_options::no_pattern, normal(), numer(), GiNaC::ex::op(), pow(), replace_with_symbol(), and GiNaC::ex::subs().

Referenced by GiNaC::basic::normal(), GiNaC::numeric::normal(), GiNaC::power::normal(), GiNaC::pseries::normal(), replace_with_symbol(), replace_with_symbol(), GiNaC::basic::to_polynomial(), GiNaC::numeric::to_polynomial(), GiNaC::power::to_polynomial(), GiNaC::basic::to_rational(), GiNaC::numeric::to_rational(), and GiNaC::power::to_rational().

7.1.3.588 replace_with_symbol() [2/2]

```
ex GiNaC::replace_with_symbol (
            const ex & e,
            exmap & repl)  [static]
```

Create a symbol for replacing the expression "e" (or return a previously assigned symbol).

The symbol and expression are appended to repl, and the symbol is returned.

See also

basic::to_rational

basic::to_polynomial

References dynallocate(), GiNaC::subs_options::no_pattern, replace_with_symbol(), and GiNaC::ex::subs().

7.1.3.589 frac_cancel()

```
ex GiNaC::frac_cancel (
            const ex & n,
            const ex & d)  [static]
```

Fraction cancellation.

Parameters

n	numerator
d	denominator

Returns

cancelled fraction {n, d} as a list

References _ex1, _ex_1, _num1_p, GiNaC::numeric::denom(), dynallocate(), ex_to(), GiNaC::ex::expand(), frac_cancel(), gcd(), get_first_symbol(), GINAC_ASSERT, GiNaC::ex::is_equal(), is_exactly_a(), is_negative(), GiNaC::ex::is_zero(), lcm_of_coefficients_denominators(), multiply_lcm(), GiNaC::numeric::numer(), and GiNaC::ex::unit().

Referenced by frac_cancel(), GiNaC::add::normal(), and GiNaC::mul::normal().

7.1.3.590 find_common_factor()

```
ex GiNaC::find_common_factor (
            const ex & e,
            ex & factor,
            exmap & repl)  [static]
```

Remove the common factor in the terms of a sum 'e' by calculating the GCD, and multiply it into the expression 'factor' (which needs to be initialized to 1, unless you're accumulating factors).

References _ex0, _ex1, divide(), dynallocate(), factor(), find_common_factor(), gcd(), GiNaC::ex::info(), GiNaC::info_flags::integer, is_a(), GiNaC::ex::is_equal(), is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::ex::nops(), GiNaC::ex::op(), pow(), and GiNaC::ex::to_polynomial().

Referenced by collect_common_factors(), and find_common_factor().

7.1.3.591 collect_common_factors()

```
ex GiNaC::collect_common_factors (
             const ex & e)
```

Collect common factors in sums.

This converts expressions like 'a∗(b∗x+b∗y)' to 'a∗b∗(x+y)'.

References collect_common_factors(), factor(), find_common_factor(), is_exactly_a(), GiNaC::subs_options::no_pattern, and GiNaC::ex::subs().

Referenced by collect_common_factors(), and GiNaC::power::to_polynomial().

7.1.3.592 resultant()

```
ex GiNaC::resultant (
             const ex & e1,
             const ex & e2,
             const ex & s)
```

Resultant of two expressions e1,e2 with respect to symbol s.

Method: Compute determinant of Sylvester matrix of e1,e2,s.

References GiNaC::ex::coeff(), GiNaC::ex::degree(), GiNaC::matrix::determinant(), GiNaC::ex::expand(), GiNaC::ex::info(), GiNaC::ex::ldegree(), GiNaC::info_flags::polynomial, and resultant().

Referenced by resultant().

7.1.3.593 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [27/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             numeric ,
             basic ,
             print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_csrc > &::do_print_csrc.  print_func< print_csrc_cl_N > &::↩
do_print_csrc_cl_N. print_func< print_tree > &::do_print_tree.  print_func< print_python_repr
> &::do_print_python_repr )
```

default ctor.

Numerically it initializes to an integer zero.

References GiNaC::numeric::do_print(), GiNaC::numeric::do_print_csrc(), GiNaC::numeric::do_print_csrc_cl_N(), GiNaC::numeric::do_print_latex(), GiNaC::numeric::do_print_python_repr(), GiNaC::numeric::do_print_tree(), GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GiNaC::basic::setflag(), and GiNaC::numeric::value.

7.1.3.594 make_real_float()

```
const cln::cl_F GiNaC::make_real_float (
            const cln::cl_idecoded_float & dec)  [static]
```

Construct a floating point number from sign, mantissa, and exponent.

References make_real_float().

Referenced by make_real_float(), and read_real_float().

7.1.3.595 read_real_float()

```
const cln::cl_F GiNaC::read_real_float (
            std::istream & s)  [static]
```

Read serialized floating point number.

References make_real_float(), and read_real_float().

Referenced by GiNaC::numeric::read_archive(), and read_real_float().

7.1.3.596 GINAC_BIND_UNARCHIVER() [37/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            numeric )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.597 write_real_float()

```
void GiNaC::write_real_float (
            std::ostream & s,
            const cln::cl_R & n)  [static]
```

References write_real_float().

Referenced by GiNaC::numeric::archive(), and write_real_float().

7.1.3.598 print_real_number()

```
void GiNaC::print_real_number (
            const print_context & c,
            const cln::cl_R & x)  [static]
```

Helper function to print a real number in a nicer way than is CLN's default.

Instead of printing 42.0L0 this just prints 42.0 to ostream os and instead of 3.99168L7 it prints 3.99168E7. This is fine in GiNaC as long as it only uses cl_LF and no other floating point types that we might want to visibly distinguish from cl_LF.

See also

numeric::print()

References is_a(), print_real_number(), and GiNaC::print_context::s.

Referenced by GiNaC::numeric::print_numeric(), print_real_cl_N(), and print_real_number().

7.1.3.599 print_integer_csrc()

```
void GiNaC::print_integer_csrc (
             const print_context & c,
             const cln::cl_I & x)  [static]
```

Helper function to print integer number in C++ source format.

See also

numeric::print()

References print_integer_csrc(), and GiNaC::print_context::s.

Referenced by print_integer_csrc(), and print_real_csrc().

7.1.3.600 print_real_csrc()

```
void GiNaC::print_real_csrc (
             const print_context & c,
             const cln::cl_R & x)  [static]
```

Helper function to print real number in C++ source format.

See also

numeric::print()

References denom(), numer(), print_integer_csrc(), print_real_csrc(), and GiNaC::print_context::s.

Referenced by GiNaC::numeric::do_print_csrc(), and print_real_csrc().

7.1.3.601 coerce()

```
template<typename T1, typename T2>
bool GiNaC::coerce (
             T1 & dst,
             const T2 & arg)  [inline], [static]
```

References coerce().

Referenced by coerce(), and print_real_cl_N().

7.1.3.602 coerce< int, cln::cl_I >()

```
template<>
bool GiNaC::coerce< int, cln::cl_I > (
             int & dst,
             const cln::cl_I & arg)  [inline]
```

Check if CLN integer can be converted into int.

See also

https://www.ginac.de/pipermail/cln-list/2006-October/000248.html

7.1.3.603 coerce< unsigned int, cln::cl_I >()

```
template<>
bool GiNaC::coerce< unsigned int, cln::cl_I > (
             unsigned int & dst,
             const cln::cl_I & arg)  [inline]
```

7.1.3.604 print_real_cl_N()

```
void GiNaC::print_real_cl_N (
             const print_context & c,
             const cln::cl_R & x)  [static]
```

Helper function to print real number in C++ source format using cl_N types.

See also

numeric::print()

References coerce(), Digits, print_real_cl_N(), print_real_number(), and GiNaC::print_context::s.

Referenced by GiNaC::numeric::do_print_csrc_cl_N(), and print_real_cl_N().

7.1.3.605 exp()

```
const numeric GiNaC::exp (
             const numeric & x)
```

Exponential function.

Returns

arbitrary precision numerical exp(x).

References exp(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by abs_eval(), abs_power(), Bernoulli_polynomial(), beta_evalf(), GiNaC::Eisenstein_h_kernel::coefficient_an(), csgn_power(), GiNaC::power::do_print_csrc(), exp(), exp_conjugate(), exp_deriv(), exp_eval(), exp_evalf(), exp_expand(), exp_imag_part(), exp_power(), exp_real_part(), generalised_Bernoulli_number(), GiNaC::Kronecker_dtau_kernel::get_ GINAC_DECLARE_UNARCHIVER(), GiNaC::power::imag_part(), log_eval(), print_sym_pow(), GiNaC::Gt::qExpand(), GiNaC::power::real_part(), REGISTER_FUNCTION(), replace_with_symbol(), GiNaC::power::series(), GiNaC::Kronecker_dtau_kerne GiNaC::Kronecker_dz_kernel::series_coeff_impl(), and tgamma().

7.1.3.606 log()

```
const numeric GiNaC::log (
            const numeric & x)
```

Natural logarithm.

Parameters

x	complex number

Returns

arbitrary precision numerical log(x).

Exceptions

pole_error("log()	logarithmic pole",0)

References GiNaC::numeric::is_zero(), log(), GiNaC::numeric::numeric(), GiNaC::pole_error::pole_error(), and GiNaC::numeric::to_cl_N().

Referenced by atan_series(), atanh_series(), GiNaC::power::derivative(), GiNaC::integral::eval_integ(), exp_eval(), G2_eval(), G2_evalf(), G3_eval(), G3_evalf(), GINAC_DECLARE_UNARCHIVER(), H_eval(), GiNaC::power::imag_part(), lgamma(), lgamma_eval(), lgamma_series(), Li2_deriv(), Li2_eval(), Li2_series(), Li_eval(), log(), log_conjugate(), log_eval(), log_evalf(), log_expand(), log_real_part(), log_series(), psi1_eval(), psi2_eval(), GiNaC::Gt::qExpand(), GiNaC::power::real_part(), REGISTER_FUNCTION(), S_eval(), GiNaC::power::series(), and GiNaC::Gt::tauToFundamental().

7.1.3.607 sin()

```
const numeric GiNaC::sin (
            const numeric & x)
```

Numeric sine (trigonometric function).

Returns

arbitrary precision numerical sin(x).

References GiNaC::numeric::numeric(), sin(), and GiNaC::numeric::to_cl_N().

Referenced by GiNaC::Eisenstein_h_kernel::coefficient_a0(), cos_deriv(), cos_imag_part(), cosh_imag_part(), exp_imag_part(), GINAC_DECLARE_UNARCHIVER(), GiNaC::power::imag_part(), lgamma(), REGISTER_FUNCTION(), sin(), sin_conjugate(), sin_eval(), sin_evalf(), sin_real_part(), sinh_eval(), sinh_imag_part(), and tan_series().

7.1.3.608 cos()

```
const numeric GiNaC::cos (
             const numeric & x)
```

Numeric cosine (trigonometric function).

Returns

arbitrary precision numerical cos(x).

References cos(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by GiNaC::Eisenstein_h_kernel::coefficient_a0(), cos(), cos_conjugate(), cos_eval(), cos_evalf(), cos_real_part(), cosh_eval(), cosh_real_part(), exp_real_part(), GINAC_DECLARE_UNARCHIVER(), GiNaC::power::real_part(), REGISTER_FUNCTION(), sin_deriv(), sin_imag_part(), sinh_real_part(), and tan_series().

7.1.3.609 tan()

```
const numeric GiNaC::tan (
             const numeric & x)
```

Numeric tangent (trigonometric function).

Returns

arbitrary precision numerical tan(x).

References GiNaC::numeric::numeric(), tan(), and GiNaC::numeric::to_cl_N().

Referenced by GINAC_DECLARE_UNARCHIVER(), REGISTER_FUNCTION(), tan(), tan_conjugate(), tan_deriv(), tan_eval(), tan_evalf(), tan_imag_part(), tan_real_part(), tanh_eval(), tanh_imag_part(), and tanh_real_part().

7.1.3.610 asin()

```
const numeric GiNaC::asin (
             const numeric & x)
```

Numeric inverse sine (trigonometric function).

Returns

arbitrary precision numerical asin(x).

References asin(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by asin(), asin_conjugate(), asin_eval(), asin_evalf(), cos_eval(), GINAC_DECLARE_UNARCHIVER(), REGISTER_FUNCTION(), sin_eval(), and tan_eval().

7.1.3.611 acos()

```
const numeric GiNaC::acos (
            const numeric & x)
```

Numeric inverse cosine (trigonometric function).

Returns

arbitrary precision numerical acos(x).

References acos(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by acos(), acos_conjugate(), acos_eval(), acos_evalf(), cos_eval(), GINAC_DECLARE_UNARCHIVER(), REGISTER_FUNCTION(), sin_eval(), and tan_eval().

7.1.3.612 atan() [1/2]

```
const numeric GiNaC::atan (
            const numeric & x)
```

Numeric arcustangent.

Parameters

x	complex number

Returns

atan(x)

Exceptions

pole_error("atan()	logarithmic pole",0) if x==I or x==-I.

References _num1_p, abs(), atan(), GiNaC::numeric::imag(), GiNaC::numeric::is_equal(), GiNaC::numeric::is_real(), GiNaC::numeric::is_zero(), GiNaC::numeric::numeric(), GiNaC::pole_error::pole_error(), GiNaC::numeric::real(), and GiNaC::numeric::to_cl_N().

Referenced by atan(), atan(), atan2_eval(), atan2_evalf(), atan_conjugate(), atan_eval(), atan_evalf(), atan_series(), cos_eval(), GINAC_DECLARE_UNARCHIVER(), REGISTER_FUNCTION(), sin_eval(), and tan_eval().

7.1.3.613 atan() [2/2]

```
const numeric GiNaC::atan (
            const numeric & y,
            const numeric & x)
```

Numeric arcustangent of two arguments, analytically continued in a suitable way.

Parameters

y	complex number
x	complex number

Returns

-I∗log((x+I∗y)/sqrt(x^2+y^2)), which is equal to atan(y/x) if y and x are both real.

Exceptions

pole_error("atan()	logarithmic pole",0) if y/x==+I or y/x==-I.

References _num0_p, atan(), GiNaC::numeric::is_real(), GiNaC::numeric::is_zero(), GiNaC::numeric::numeric(), GiNaC::pole_error::pole_error(), and GiNaC::numeric::to_cl_N().

7.1.3.614 sinh()

```
const numeric GiNaC::sinh (
            const numeric & x)
```

Numeric hyperbolic sine (trigonometric function).

Returns

arbitrary precision numerical sinh(x).

References GiNaC::numeric::numeric(), sinh(), and GiNaC::numeric::to_cl_N().

Referenced by cos_imag_part(), cosh_deriv(), cosh_imag_part(), GINAC_DECLARE_UNARCHIVER(), REGISTER_FUNCTION(), sin_imag_part(), sinh(), sinh_conjugate(), sinh_eval(), sinh_evalf(), sinh_real_part(), and tanh_series().

7.1.3.615 cosh()

```
const numeric GiNaC::cosh (
            const numeric & x)
```

Numeric hyperbolic cosine (trigonometric function).

Returns

arbitrary precision numerical cosh(x).

References cosh(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by cos_real_part(), cosh(), cosh_conjugate(), cosh_eval(), cosh_evalf(), cosh_real_part(), GINAC_DECLARE_UNARCHIVER(), REGISTER_FUNCTION(), sin_real_part(), sinh_deriv(), sinh_imag_part(), and tanh_series().

7.1.3.616 tanh()

```
const numeric GiNaC::tanh (
            const numeric & x)
```

Numeric hyperbolic tangent (trigonometric function).

Returns

arbitrary precision numerical tanh(x).

References GiNaC::numeric::numeric(), tanh(), and GiNaC::numeric::to_cl_N().

Referenced by GINAC_DECLARE_UNARCHIVER(), REGISTER_FUNCTION(), tan_imag_part(), tanh(), tanh_conjugate(), tanh_deriv(), tanh_eval(), tanh_evalf(), tanh_imag_part(), and tanh_real_part().

7.1.3.617 asinh()

```
const numeric GiNaC::asinh (
            const numeric & x)
```

Numeric inverse hyperbolic sine (trigonometric function).

Returns

arbitrary precision numerical asinh(x).

References asinh(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by asinh(), asinh_conjugate(), asinh_eval(), asinh_evalf(), cosh_eval(), GINAC_DECLARE_UNARCHIVER(), REGISTER_FUNCTION(), sinh_eval(), and tanh_eval().

7.1.3.618 acosh()

```
const numeric GiNaC::acosh (
            const numeric & x)
```

Numeric inverse hyperbolic cosine (trigonometric function).

Returns

arbitrary precision numerical acosh(x).

References acosh(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by acosh(), acosh_conjugate(), acosh_eval(), acosh_evalf(), cosh_eval(), GINAC_DECLARE_UNARCHIVER(), REGISTER_FUNCTION(), sinh_eval(), and tanh_eval().

7.1.3.619 atanh()

```
const numeric GiNaC::atanh (
            const numeric & x)
```

Numeric inverse hyperbolic tangent (trigonometric function).

Returns

arbitrary precision numerical atanh(x).

References atanh(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by atanh(), atanh_conjugate(), atanh_eval(), atanh_evalf(), atanh_series(), cosh_eval(), GINAC_DECLARE_UNARCHIVE REGISTER_FUNCTION(), sinh_eval(), and tanh_eval().

7.1.3.620 Li2_series() [2/2]

```
cln::cl_N GiNaC::Li2_series (
            const cln::cl_N & x,
            const cln::float_format_t & prec)  [static]
```

Numeric evaluation of Dilogarithm within circle of convergence (unit circle) using a power series.

References Li2_series().

7.1.3.621 Li2_projection()

```
cln::cl_N GiNaC::Li2_projection (
            const cln::cl_N & x,
            const cln::float_format_t & prec)  [static]
```

Folds Li2's argument inside a small rectangle to enhance convergence.

References Li2_projection(), and Li2_series().

Referenced by Li2_(), and Li2_projection().

7.1.3.622 Li2_()

```
const cln::cl_N GiNaC::Li2_ (
            const cln::cl_N & value)
```

Numeric evaluation of Dilogarithm.

The domain is the entire complex plane, the branch cut lies along the positive real axis, starting at 1 and continuous with quadrant IV.

Returns

arbitrary precision numerical Li2(x).

References Li2_(), and Li2_projection().

Referenced by Li2(), and Li2_().

7.1.3.623 Li2()

```
const numeric GiNaC::Li2 (
            const numeric & x)
```

References _num0_p, Li2(), Li2_(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by GINAC_DECLARE_UNARCHIVER(), Li2(), Li2_conjugate(), Li2_eval(), Li2_evalf(), Li2_series(), and REGISTER_FUNCTION().

7.1.3.624 zeta() [3/3]

```
const numeric GiNaC::zeta (
            const numeric & x)
```

Numeric evaluation of Riemann's Zeta function.

Currently works only for integer arguments.

References GiNaC::numeric::is_real(), GiNaC::numeric::numeric(), GiNaC::numeric::to_cl_N(), and zeta().

7.1.3.625 guess_precision()

```
cln::float_format_t GiNaC::guess_precision (
            const cln::cl_N & x)  [static]
```

References guess_precision().

Referenced by guess_precision(), lgamma(), and tgamma().

7.1.3.626 lgamma() [1/2]

```
const cln::cl_N GiNaC::lgamma (
            const cln::cl_N & x)
```

The Gamma function.

Use the Lanczos approximation. If the coefficients used here are not sufficiently many or sufficiently accurate, more can be calculated using the program doc/examples/lanczos.cpp. In that case, be sure to read the comments in that file.

References GiNaC::lanczos_coeffs::calc_lanczos_A(), GiNaC::lanczos_coeffs::get_order(), guess_precision(), lgamma(), log(), sin(), and GiNaC::lanczos_coeffs::sufficiently_accurate().

Referenced by beta_evalf(), GINAC_DECLARE_UNARCHIVER(), lgamma(), lgamma(), lgamma_conjugate(), lgamma_eval(), lgamma_evalf(), lgamma_series(), and REGISTER_FUNCTION().

7.1.3.627 lgamma() [2/2]

```
const numeric GiNaC::lgamma (
            const numeric & x)
```

References lgamma(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

7.1.3.628 tgamma() [1/2]

```
const cln::cl_N GiNaC::tgamma (
             const cln::cl_N & x)
```

References GiNaC::lanczos_coeffs::calc_lanczos_A(), exp(), GiNaC::lanczos_coeffs::get_order(), guess_precision(), sqrt(), GiNaC::lanczos_coeffs::sufficiently_accurate(), and tgamma().

Referenced by beta_eval(), beta_series(), GINAC_DECLARE_UNARCHIVER(), psi2_eval(), REGISTER_FUNCTION(), tgamma(), tgamma(), tgamma_conjugate(), tgamma_deriv(), tgamma_eval(), tgamma_evalf(), and tgamma_series().

7.1.3.629 tgamma() [2/2]

```
const numeric GiNaC::tgamma (
             const numeric & x)
```

References GiNaC::numeric::numeric(), tgamma(), and GiNaC::numeric::to_cl_N().

7.1.3.630 psi() [3/4]

```
const numeric GiNaC::psi (
             const numeric & x)
```

The psi function (aka polygamma function).

This is only a stub!

References psi().

7.1.3.631 psi() [4/4]

```
const numeric GiNaC::psi (
             const numeric & n,
             const numeric & x)
```

The psi functions (aka polygamma functions).

This is only a stub!

References psi().

7.1.3.632 factorial()

```
const numeric GiNaC::factorial (
            const numeric & n)
```

Factorial combinatorial function.

Parameters

n	integer argument >= 0

Exceptions

range_error	(argument must be integer >= 0)

References factorial(), GiNaC::numeric::is_nonneg_integer(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_int().

Referenced by Bernoulli_polynomial(), GiNaC::Gt::eval(), factorial(), factorial_conjugate(), factorial_eval(), factorial_evalf(), factorial_real_part(), G2_eval(), G2_evalf(), G3_eval(), G3_evalf(), GiNaC::Gt_detail::pathintegral_term::G_path(), generalised_Bernoulli_number(), GINAC_DECLARE_UNARCHIVER(), H_eval(), lgamma_eval(), multinomial_coefficient(), psi2_eval(), psi2_series(), GiNaC::Gt::qExpand(), REGISTER_FUNCTION(), GiNaC::Gt::regularise(), S_eval(), GiNaC::Kronecker_dtau_kernel::series_coeff_impl(), GiNaC::Kronecker_dz_kernel::series_coeff_impl(), symm(), GiNaC::Gt::tauToFundamental(), tgamma_eval(), zeta1_eval(), and GiNaC::Gt::zisToFundamental().

7.1.3.633 doublefactorial()

```
const numeric GiNaC::doublefactorial (
            const numeric & n)
```

The double factorial combinatorial function.

(Scarcely used, but still useful in cases, like for exact results of tgamma(n+1/2) for instance.)

Parameters

n	integer argument >= -1

Returns

n!! == n * (n-2) * (n-4) * ... * ({1|2}) with 0!! == (-1)!! == 1

Exceptions

range_error	(argument must be integer >= -1)

References _num1_p, _num_1_p, doublefactorial(), GiNaC::numeric::is_equal(), GiNaC::numeric::is_nonneg_integer(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_int().

Referenced by doublefactorial(), GINAC_DECLARE_UNARCHIVER(), and tgamma_eval().

7.1.3.634 binomial()

```
const numeric GiNaC::binomial (
             const numeric & n,
             const numeric & k)
```

The Binomial coefficients.

It computes the binomial coefficients. For integer n and k and positive n this is the number of ways of choosing k objects from n distinct objects. If n is a negative integer, the formula binomial(n,k) == (-1)^k∗binomial(k-n-1,k) (if k>=0) binomial(n,k) == (-1)^(n-k)∗binomial(-k-1,n-k) (otherwise) is used to compute the result.

References _num0_p, _num1_p, _num_1_p, binomial(), GiNaC::numeric::compare(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_nonneg_integer(), GiNaC::numeric::numeric(), GiNaC::numeric::power(), and GiNaC::numeric::to_int().

Referenced by binomial(), binomial_conjugate(), binomial_eval(), binomial_evalf(), binomial_real_part(), binomial_sym(), GiNaC::power::expand_add(), GINAC_DECLARE_UNARCHIVER(), GiNaC::power::imag_part(), GiNaC::Gt::qExpand(), GiNaC::power::real_part(), REGISTER_FUNCTION(), and GiNaC::Gt::tauToFundamental().

7.1.3.635 bernoulli()

```
const numeric GiNaC::bernoulli (
             const numeric & nn)
```

Bernoulli number.

The nth Bernoulli number is the coefficient of x^n/n! in the expansion of the function x/(e^x-1).

Returns

the nth Bernoulli number (a rational number).

Exceptions

range_error	(argument must be integer >= 0)

References _num1_p, bernoulli(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_negative(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_int().

Referenced by bernoulli(), GINAC_DECLARE_UNARCHIVER(), GiNaC::Kronecker_dtau_kernel::series_coeff_impl(), GiNaC::Kronecker_dz_kernel::series_coeff_impl(), and zeta1_eval().

7.1.3.636 fibonacci()

```
const numeric GiNaC::fibonacci (
              const numeric & n)
```

Fibonacci number.

The nth Fibonacci number F(n) is defined by the recurrence formula F(n)==F(n-1)+F(n-2) with F(0)==0 and F(1)==1.

Parameters

n	an integer

Returns

the nth Fibonacci number F(n) (an integer number)

Exceptions

range_error	(argument must be an integer)

References _num0_p, fibonacci(), GiNaC::numeric::is_even(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_negative(), GiNaC::numeric::is_zero(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by fibonacci(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.637 abs()

```
const numeric GiNaC::abs (
              const numeric & x)
```

Absolute value.

References abs(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by abs(), abs_conjugate(), abs_eval(), abs_evalf(), abs_expand(), abs_expl_derivative(), abs_power(), abs_real_part(), adaptivesimpson(), atan(), atan_series(), atanh_series(), GiNaC::power::eval(), GiNaC::Gt::findMoebiusTransform(), generalised_Bernoulli_number(), GINAC_DECLARE_UNARCHIVER(), GiNaC::power::imag_part(), GiNaC::mul::integer_content(), GiNaC::numeric::integer_content(), is_discriminant_of_quadratic_number_field(), log_real_part(), GiNaC::add::max_coefficient(), GiNaC::mul::max_coefficient(), GiNaC::numeric::max_coefficient(), GiNaC::matrix::pivot(), GiNaC::Gt::qExpand(), GiNaC::power::real_part(), REGISTER_FUNCTION(), tgamma_eval(), GiNaC::ex::unitcontprim(), and zeta1_eval().

7.1.3.638 mod()

```
const numeric GiNaC::mod (
              const numeric & a,
              const numeric & b)
```

Modulus (in positive representation).

In general, mod(a,b) has the sign of b or is zero, and rem(a,b) has the sign of a or is zero. This is different from Maple's modp, where the sign of b is ignored. It is in agreement with Mathematica's Mod.

Returns

a mod b in the range [0,abs(b)-1] with sign of b if both are integer, 0 otherwise.

References _num0_p, GiNaC::numeric::is_integer(), mod(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by GiNaC::Eisenstein_h_kernel::coefficient_a0(), GiNaC::Eisenstein_h_kernel::coefficient_an(), cos_eval(), exp_eval(), GINAC_DECLARE_UNARCHIVER(), is_discriminant_of_quadratic_number_field(), mod(), sin_eval(), and tan_eval().

7.1.3.639 smod()

```
const numeric GiNaC::smod (
             const numeric & a_,
             const numeric & b_)
```

Modulus (in symmetric representation).

Returns

a mod b in the range [-iquo(abs(b),2), iquo(abs(b),2)].

References _num0_p, GiNaC::numeric::is_integer(), GiNaC::numeric::numeric(), smod(), and GiNaC::numeric::to_cl_N().

Referenced by GINAC_DECLARE_UNARCHIVER(), GiNaC::add::smod(), GiNaC::mul::smod(), GiNaC::numeric::smod(), and smod().

7.1.3.640 irem() [1/2]

```
const numeric GiNaC::irem (
             const numeric & a,
             const numeric & b)
```

Numeric integer remainder.

Equivalent to Maple's irem(a,b) as far as sign conventions are concerned. In general, mod(a,b) has the sign of b or is zero, and irem(a,b) has the sign of a or is zero.

Returns

remainder of a/b if both are integer, 0 otherwise.

Exceptions

overflow_error	(division by zero) if b is zero.

References _num0_p, irem(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_zero(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by GiNaC::Eisenstein_h_kernel::coefficient_a0(), GiNaC::Eisenstein_h_kernel::coefficient_an(), GINAC_DECLARE_UNARCHIVER(), ifactor(), irem(), and irem().

7.1.3.641 irem() [2/2]

```
const numeric GiNaC::irem (
              const numeric & a,
              const numeric & b,
              numeric & q)
```

Numeric integer remainder.

Equivalent to Maple's irem(a,b,'q') it obeyes the relation irem(a,b,q) == a - q∗b. In general, mod(a,b) has the sign of b or is zero, and irem(a,b) has the sign of a or is zero.

Returns

remainder of a/b and quotient stored in q if both are integer, 0 otherwise.

Exceptions

overflow_error	(division by zero) if b is zero.

References _num0_p, irem(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_zero(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

7.1.3.642 iquo() [1/2]

```
const numeric GiNaC::iquo (
              const numeric & a,
              const numeric & b)
```

Numeric integer quotient.

Equivalent to Maple's iquo as far as sign conventions are concerned.

Returns

truncated quotient of a/b if both are integer, 0 otherwise.

Exceptions

overflow_error	(division by zero) if b is zero.

References _num0_p, iquo(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_zero(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by GiNaC::power::eval(), GINAC_DECLARE_UNARCHIVER(), heur_gcd_z(), iquo(), and iquo().

7.1.3.643 iquo() [2/2]

```
const numeric GiNaC::iquo (
             const numeric & a,
             const numeric & b,
             numeric & r)
```

Numeric integer quotient.

Equivalent to Maple's iquo(a,b,'r') it obeyes the relation r == a - iquo(a,b,r)∗b.

Returns

truncated quotient of a/b and remainder stored in r if both are integer, 0 otherwise.

Exceptions

overflow_error	(division by zero) if b is zero.

References _num0_p, iquo(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_zero(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

7.1.3.644 gcd() [2/2]

```
const numeric GiNaC::gcd (
             const numeric & a,
             const numeric & b)
```

Greatest Common Divisor.

Returns

The GCD of two numbers if both are integer, a numerical 1 if they are not.

References _num1_p, gcd(), GiNaC::numeric::is_integer(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

7.1.3.645 lcm() [2/2]

```
const numeric GiNaC::lcm (
             const numeric & a,
             const numeric & b)
```

Least Common Multiple.

Returns

The LCM of two numbers if both are integer, the product of those two numbers if they are not.

References GiNaC::numeric::is_integer(), lcm(), GiNaC::numeric::mul(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

7.1.3.646 sqrt() [1/2]

```
const numeric GiNaC::sqrt (
            const numeric & x)
```

Numeric square root.

If possible, sqrt(x) should respect squares of exact numbers, i.e. sqrt(4) should return integer 2.

Parameters

x	numeric argument

Returns

square root of x. Branch cut along negative real axis, the negative real axis itself where imag(x)==0 and real(x)<0 belongs to the upper part where imag(x)>0.

References GiNaC::numeric::numeric(), sqrt(), and GiNaC::numeric::to_cl_N().

Referenced by clifford_norm(), cos_eval(), cosh_eval(), EllipticE_evalf(), EllipticK_evalf(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GINAC_DECLARE_UNARCHIVER(), sin_eval(), sinh_eval(), sqrt(), sqrt(), tan_eval(), tanh_eval(), tgamma(), and tgamma_eval().

7.1.3.647 isqrt()

```
const numeric GiNaC::isqrt (
            const numeric & x)
```

Integer numeric square root.

References _num0_p, GiNaC::numeric::is_integer(), isqrt(), GiNaC::numeric::numeric(), and GiNaC::numeric::to_cl_N().

Referenced by GINAC_DECLARE_UNARCHIVER(), heur_gcd_z(), and isqrt().

7.1.3.648 PiEvalf()

```
ex GiNaC::PiEvalf ()
```

Floating point evaluation of Archimedes' constant Pi.

References GiNaC::numeric::numeric(), and PiEvalf().

Referenced by PiEvalf().

7.1.3.649 EulerEvalf()

```
ex GiNaC::EulerEvalf ()
```

Floating point evaluation of Euler's constant gamma.

References EulerEvalf(), and GiNaC::numeric::numeric().

Referenced by EulerEvalf().

7.1.3.650 CatalanEvalf()

```
ex GiNaC::CatalanEvalf ()
```

Floating point evaluation of Catalan's constant.

References CatalanEvalf(), and GiNaC::numeric::numeric().

Referenced by CatalanEvalf().

7.1.3.651 operator<<() [6/16]

```
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const _numeric_digits & e)
```

References operator<<(), and GiNaC::_numeric_digits::print().

7.1.3.652 GINAC_DECLARE_UNARCHIVER() [39/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            numeric )
```

References abs(), acos(), acosh(), asin(), asinh(), atan(), atanh(), bernoulli(), binomial(), cos(), cosh(), Digits, doublefactorial(), exp(), factorial(), fibonacci(), gcd(), GINAC_DECLARE_UNARCHIVER(), I, iquo(), irem(), isqrt(), lcm(), lgamma(), Li2(), log(), mod(), psi(), sin(), sinh(), smod(), sqrt(), tan(), tanh(), tgamma(), and zeta().

7.1.3.653 pow() [1/3]

```
const numeric GiNaC::pow (
            const numeric & x,
            const numeric & y)  [inline]
```

References pow(), and GiNaC::numeric::power().

Referenced by abs_eval(), abs_power(), GiNaC::mul::algebraic_subs_mul(), beta_eval(), clifford_inverse(), clifford_prime(), clifford_star_bar(), clifford_to_lst(), GiNaC::Eisenstein_h_kernel::coefficient_an(), GiNaC::basic::collect(), GiNaC::mul::combine_ex_with_coeff_to_pair(), GiNaC::mul::combine_pair_with_coeff_to_pair(), GiNaC::pseries::convert_to_poly(), GiNaC::mul::derivative(), GiNaC::power::derivative(), divide(), divide_in_z(), EllipticE_series(), EllipticK_series(), GiNaC::Gt::eval(), GiNaC::power::eval(), GiNaC::power::evalm(), GiNaC::Gt_detail::pathintegral_term::evaluate(), GiNaC::power::expand(), GiNaC::power::expand_add(), factor(), find_common_factor(), G2_eval(), G2_evalf(), G3_eval(), G3_evalf(), GiNaC::Gt_detail::pathintegral_term::G_path(), gcd(), gcd_pf_pow(), gcd_pf_pow_pow(), GiNaC::Kronecker_dtau_kernel::get_numerical_value(), GiNaC::Kronecker_dz_kernel::get_numerical_value(), H_eval(), GiNaC::power::imag_part(), GiNaC::Gt_detail::pathintegral_term::integrate(), interpolate(), kronecker_symbol(), GiNaC::integration_kernel::Laurent_series(), lcmcoeff(), Li2_series(), Li_eval(), Li_series(), log_series(), multiply_lcm(), GiNaC::power::normal(), GiNaC::pseries::op(), Order_power(), pow(), pow(), pow(), GiNaC::pseries::power_const(), prem(), GiNaC::pseries::print_series(), psi1_eval(), psi2_eval(), GiNaC::Eisenstein_h_kernel::q_expansion_modular_form(), GiNaC::Gt::qExpand(), quo(), GiNaC::power::real_part(), GiNaC::Gt::regularise(), rem(), remove_dirac_ONE(), replace_with_symbol(), S_eval(), S_series(), GiNaC::modular_form_kernel::series(), GiNaC::Kronecker_dz_kernel::series_coeff_impl sprem(), sqrfree_parfrac(), sr_gcd(), GiNaC::power::subs(), GiNaC::Gt::tauToFundamental(), tgamma_eval(), GiNaC::power::to_polynomial(), GiNaC::power::to_rational(), zeta1_eval(), and GiNaC::Gt::zisToFundamental().

7.1.3.654 inverse() [3/3]

```
const numeric GiNaC::inverse (
            const numeric & x)  [inline]
```

References inverse(), and GiNaC::numeric::inverse().

7.1.3.655 step()

```
numeric GiNaC::step (
            const numeric & x)  [inline]
```

References GiNaC::numeric::step(), and step().

Referenced by abs_eval(), H_eval(), REGISTER_FUNCTION(), step(), step_conjugate(), step_eval(), step_evalf(), step_real_part(), and step_series().

7.1.3.656 csgn()

```
int GiNaC::csgn (
            const numeric & x)  [inline]
```

References csgn(), and GiNaC::numeric::csgn().

Referenced by atan_series(), atanh_series(), csgn(), csgn_conjugate(), csgn_eval(), csgn_evalf(), csgn_power(), csgn_real_part(), csgn_series(), eta_eval(), eta_evalf(), log_series(), and REGISTER_FUNCTION().

7.1.3.657 is_zero() [2/2]

```
bool GiNaC::is_zero (
            const numeric & x)  [inline]
```

References is_zero(), and GiNaC::numeric::is_zero().

7.1.3.658 is_positive()

```
bool GiNaC::is_positive (
            const numeric & x)  [inline]
```

References is_positive(), and GiNaC::numeric::is_positive().

Referenced by beta_eval(), and is_positive().

7.1.3.659 is_negative()

```
bool GiNaC::is_negative (
            const numeric & x)  [inline]
```

References is_negative(), and GiNaC::numeric::is_negative().

Referenced by GiNaC::power::do_print_latex(), GiNaC::power::eval(), frac_cancel(), and is_negative().

7.1.3.660 is_integer()

```
bool GiNaC::is_integer (
            const numeric & x)  [inline]
```

References is_integer(), and GiNaC::numeric::is_integer().

Referenced by beta_eval(), binomial_eval(), GiNaC::power::coeff(), GiNaC::power::degree(), GiNaC::power::expand(), is_integer(), GiNaC::power::ldegree(), GiNaC::pseries::power_const(), psi1_eval(), psi2_eval(), replace_with_symbol(), and GiNaC::power::series().

7.1.3.661 is_pos_integer()

```
bool GiNaC::is_pos_integer (
            const numeric & x)  [inline]
```

References is_pos_integer(), and GiNaC::numeric::is_pos_integer().

Referenced by GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), and is_pos_integer().

7.1.3.662 is_nonneg_integer()

```
bool GiNaC::is_nonneg_integer (
            const numeric & x)  [inline]
```

References is_nonneg_integer(), and GiNaC::numeric::is_nonneg_integer().

Referenced by is_nonneg_integer().

7.1.3.663 is_even()

```
bool GiNaC::is_even (
            const numeric & x)  [inline]
```

References is_even(), and GiNaC::numeric::is_even().

Referenced by abs_power(), is_even(), and GiNaC::Kronecker_dz_kernel::series_coeff_impl().

7.1.3.664 is_odd()

```
bool GiNaC::is_odd (
            const numeric & x)  [inline]
```

References is_odd(), and GiNaC::numeric::is_odd().

Referenced by csgn_power(), and is_odd().

7.1.3.665 is_prime()

```
bool GiNaC::is_prime (
            const numeric & x)  [inline]
```

References is_prime(), and GiNaC::numeric::is_prime().

Referenced by is_prime().

7.1.3.666 is_rational()

```
bool GiNaC::is_rational (
            const numeric & x)  [inline]
```

References is_rational(), and GiNaC::numeric::is_rational().

Referenced by beta_eval(), is_rational(), lgamma_eval(), replace_with_symbol(), and tgamma_eval().

7.1.3.667 is_real()

```
bool GiNaC::is_real (
            const numeric & x)  [inline]
```

References is_real(), and GiNaC::numeric::is_real().

Referenced by beta_eval(), fsolve(), G2_eval(), G2_evalf(), is_real(), and Li2_series().

7.1.3.668 is_cinteger()

```
bool GiNaC::is_cinteger (
            const numeric & x)  [inline]
```

References is_cinteger(), and GiNaC::numeric::is_cinteger().

Referenced by is_cinteger().

7.1.3.669 is_crational()

```
bool GiNaC::is_crational (
            const numeric & x)  [inline]
```

References is_crational(), and GiNaC::numeric::is_crational().

Referenced by is_crational().

7.1.3.670 to_int()

```
int GiNaC::to_int (
            const numeric & x)  [inline]
```

References GiNaC::numeric::to_int(), and to_int().

Referenced by color_trace(), dirac_trace(), H_evalf(), GiNaC::basic::operator[](), GiNaC::basic::operator[](), GiNaC::pseries::power_const(), S_eval(), GiNaC::power::series(), sqrfree_parfrac(), and to_int().

7.1.3.671 to_long()

```
long GiNaC::to_long (
            const numeric & x)  [inline]
```

References GiNaC::numeric::to_long(), and to_long().

Referenced by to_long().

7.1.3.672 to_double()

```
double GiNaC::to_double (
            const numeric & x)  [inline]
```

References GiNaC::numeric::to_double(), and to_double().

Referenced by to_double().

7.1.3.673 real()

```
const numeric GiNaC::real (
            const numeric & x)  [inline]
```

References GiNaC::numeric::real(), and real().

Referenced by cosh_eval(), GiNaC::Gt::qExpand(), real(), sinh_eval(), and tanh_eval().

7.1.3.674 imag()

```
const numeric GiNaC::imag (
            const numeric & x)  [inline]
```

References imag(), and GiNaC::numeric::imag().

Referenced by eta_eval(), eta_evalf(), G2_eval(), G2_evalf(), and imag().

7.1.3.675 numer() [2/2]

```
const numeric GiNaC::numer (
            const numeric & x)  [inline]
```

References numer(), and GiNaC::numeric::numer().

7.1.3.676 denom() [2/2]

```
const numeric GiNaC::denom (
            const numeric & x)  [inline]
```

References denom(), and GiNaC::numeric::denom().

7.1.3.677 exadd()

```
const ex GiNaC::exadd (
            const ex & lh,
            const ex & rh)  [inline], [static]
```

Used internally by operator+() to add two ex objects.

References dynallocate(), and exadd().

Referenced by exadd(), operator+(), operator++(), operator++(), operator+=(), operator-(), operator--(), operator--(), and operator-=().

7.1.3.678 exmul()

```
const ex GiNaC::exmul (
            const ex & lh,
            const ex & rh)  [inline], [static]
```

Used internally by operator∗() to multiply two ex objects.

References GiNaC::return_types::commutative, dynallocate(), exmul(), and GiNaC::ex::return_type().

Referenced by exmul(), operator∗(), operator∗=(), operator/(), and operator/=().

7.1.3.679 exminus()

```
const ex GiNaC::exminus (
            const ex & lh)  [inline], [static]
```

Used internally by operator-() and friends to change the sign of an argument.

References _ex_1, dynallocate(), and exminus().

Referenced by exminus(), operator-(), operator-(), and operator-=().

7.1.3.680 operator+() [1/4]

```
const ex GiNaC::operator+ (
             const ex & lh,
             const ex & rh)
```

References exadd(), and operator+().

Referenced by operator+(), operator+(), operator+(), and operator+().

7.1.3.681 operator-() [1/4]

```
const ex GiNaC::operator- (
             const ex & lh,
             const ex & rh)
```

References exadd(), exminus(), and operator-().

Referenced by operator-(), operator-(), operator-(), and operator-().

7.1.3.682 operator∗() [1/2]

```
const ex GiNaC::operator* (
             const ex & lh,
             const ex & rh)
```

References exmul(), and operator∗().

Referenced by operator∗(), and operator∗().

7.1.3.683 operator/() [1/2]

```
const ex GiNaC::operator/ (
             const ex & lh,
             const ex & rh)
```

References _ex_1, exmul(), operator/(), and GiNaC::power::power().

Referenced by operator/(), and operator/().

7.1.3.684 operator+() [2/4]

```
const numeric GiNaC::operator+ (
             const numeric & lh,
             const numeric & rh)
```

References GiNaC::numeric::add(), and operator+().

7.1.3.685 operator-() [2/4]

```
const numeric GiNaC::operator- (
             const numeric & lh,
             const numeric & rh)
```

References operator-(), and GiNaC::numeric::sub().

7.1.3.686 operator∗() [2/2]

```
const numeric GiNaC::operator* (
             const numeric & lh,
             const numeric & rh)
```

References GiNaC::numeric::mul(), and operator∗().

7.1.3.687 operator/() [2/2]

```
const numeric GiNaC::operator/ (
             const numeric & lh,
             const numeric & rh)
```

References GiNaC::numeric::div(), and operator/().

7.1.3.688 operator+=() [1/2]

```
ex & GiNaC::operator+= (
             ex & lh,
             const ex & rh)
```

References exadd(), and operator+=().

Referenced by operator+=(), and operator+=().

7.1.3.689 operator-=() [1/2]

```
ex & GiNaC::operator-= (
             ex & lh,
             const ex & rh)
```

References exadd(), exminus(), and operator-=().

Referenced by operator-=(), and operator-=().

7.1.3.690 operator∗=() [1/2]

```
ex & GiNaC::operator*= (
             ex & lh,
             const ex & rh)
```

References exmul(), and operator∗=().

Referenced by operator∗=(), and operator∗=().

7.1.3.691 operator/=() [1/2]

```
ex & GiNaC::operator/= (
             ex & lh,
             const ex & rh)
```

References _ex_1, exmul(), operator/=(), and GiNaC::power::power().

Referenced by operator/=(), and operator/=().

7.1.3.692 operator+=() [2/2]

```
numeric & GiNaC::operator+= (
             numeric & lh,
             const numeric & rh)
```

References GiNaC::numeric::add(), and operator+=().

7.1.3.693 operator-=() [2/2]

```
numeric & GiNaC::operator-= (
             numeric & lh,
             const numeric & rh)
```

References operator-=(), and GiNaC::numeric::sub().

7.1.3.694 operator*=() [2/2]

```
numeric & GiNaC::operator*= (
             numeric & lh,
             const numeric & rh)
```

References GiNaC::numeric::mul(), and operator*=().

7.1.3.695 operator/=() [2/2]

```
numeric & GiNaC::operator/= (
             numeric & lh,
             const numeric & rh)
```

References GiNaC::numeric::div(), and operator/=().

7.1.3.696 operator+() [3/4]

```
const ex GiNaC::operator+ (
             const ex & lh)
```

References operator+().

7.1.3.697 operator-() [3/4]

```
const ex GiNaC::operator- (
             const ex & lh)
```

References exminus(), and operator-().

7.1.3.698 operator+() [4/4]

```
const numeric GiNaC::operator+ (
             const numeric & lh)
```

References operator+().

7.1.3.699 operator-() [4/4]

```
const numeric GiNaC::operator- (
             const numeric & lh)
```

References _num_1_p, GiNaC::numeric::mul(), and operator-().

7.1.3.700 operator++() [1/4]

```
ex & GiNaC::operator++ (
             ex & rh)
```

Expression prefix increment.

Adds 1 and returns incremented ex.

References _ex1, exadd(), and operator++().

Referenced by operator++(), operator++(), operator++(), and operator++().

7.1.3.701 operator--() [1/4]

```
ex & GiNaC::operator-- (
             ex & rh)
```

Expression prefix decrement.

Subtracts 1 and returns decremented ex.

References _ex_1, exadd(), and operator--().

Referenced by operator--(), operator--(), operator--(), and operator--().

7.1.3.702 operator++() [2/4]

```
const ex GiNaC::operator++ (
             ex & lh,
             int )
```

Expression postfix increment.

Returns the ex and leaves the original incremented by 1.

References _ex1, exadd(), and operator++().

7.1.3.703 operator--() [2/4]

```
const ex GiNaC::operator-- (
             ex & lh,
             int )
```

Expression postfix decrement.

Returns the ex and leaves the original decremented by 1.

References _ex_1, exadd(), and operator--().

7.1.3.704 operator++() [3/4]

```
numeric & GiNaC::operator++ (
             numeric & rh)
```

Numeric prefix increment.

Adds 1 and returns incremented number.

References _num1_p, GiNaC::numeric::add(), and operator++().

7.1.3.705 operator--() [3/4]

```
numeric & GiNaC::operator-- (
             numeric & rh)
```

Numeric prefix decrement.

Subtracts 1 and returns decremented number.

References _num_1_p, GiNaC::numeric::add(), and operator--().

7.1.3.706 operator++() [4/4]

```
const numeric GiNaC::operator++ (
             numeric & lh,
             int )
```

Numeric postfix increment.

Returns the number and leaves the original incremented by 1.

References _num1_p, GiNaC::numeric::add(), and operator++().

7.1.3.707 operator--() [4/4]

```
const numeric GiNaC::operator-- (
             numeric & lh,
             int )
```

Numeric postfix decrement.

Returns the number and leaves the original decremented by 1.

References _num_1_p, GiNaC::numeric::add(), and operator--().

7.1.3.708 operator==()

```
const relational GiNaC::operator== (
             const ex & lh,
             const ex & rh)
```

References GiNaC::relational::equal, operator==(), and GiNaC::relational::relational().

Referenced by operator==().

7.1.3.709 operator"!=()

```
const relational GiNaC::operator!= (
             const ex & lh,
             const ex & rh)
```

References GiNaC::relational::not_equal, operator!=(), and GiNaC::relational::relational().

Referenced by operator!=().

7.1.3.710 operator<()

```
const relational GiNaC::operator< (
             const ex & lh,
             const ex & rh)
```

References GiNaC::relational::less, operator<(), and GiNaC::relational::relational().

Referenced by operator<().

7.1.3.711 operator<=()

```
const relational GiNaC::operator<= (
            const ex & lh,
            const ex & rh)
```

References GiNaC::relational::less_or_equal, operator<=(), and GiNaC::relational::relational().

Referenced by operator<=().

7.1.3.712 operator>()

```
const relational GiNaC::operator> (
            const ex & lh,
            const ex & rh)
```

References GiNaC::relational::greater, operator>(), and GiNaC::relational::relational().

Referenced by operator>().

7.1.3.713 operator>=()

```
const relational GiNaC::operator>= (
            const ex & lh,
            const ex & rh)
```

References GiNaC::relational::greater_or_equal, operator>=(), and GiNaC::relational::relational().

Referenced by operator>=().

7.1.3.714 my_ios_index()

```
int GiNaC::my_ios_index ()  [static]
```

References my_ios_index().

Referenced by get_print_context(), my_ios_index(), and set_print_context().

7.1.3.715 my_ios_callback()

```
void GiNaC::my_ios_callback (
            std::ios_base::event ev,
            std::ios_base & s,
            int i)  [static]
```

References my_ios_callback().

Referenced by my_ios_callback(), and set_print_context().

7.1.3.716 get_print_context()

```
print_context * GiNaC::get_print_context (
            std::ios_base & s)  [inline], [static]
```

References get_print_context(), and my_ios_index().

Referenced by get_print_context(), get_print_options(), operator<<(), operator<<(), operator<<(), operator<<(), and set_print_options().

7.1.3.717 set_print_context()

```
void GiNaC::set_print_context (
            std::ios_base & s,
            const print_context & c)  [static]
```

References callback_registered, my_ios_callback(), my_ios_index(), GiNaC::print_context::options, and set_print_context().

Referenced by csrc(), csrc_cl_N(), csrc_double(), csrc_float(), dflt(), latex(), python(), python_repr(), set_print_context(), set_print_options(), and tree().

7.1.3.718 get_print_options()

```
unsigned GiNaC::get_print_options (
            std::ios_base & s)  [inline], [static]
```

References get_print_context(), get_print_options(), and GiNaC::print_context::options.

Referenced by get_print_options(), index_dimensions(), and no_index_dimensions().

7.1.3.719 set_print_options()

```
void GiNaC::set_print_options (
            std::ostream & s,
            unsigned options)  [static]
```

References get_print_context(), GiNaC::print_context::options, GiNaC::print_dflt::print_dflt(), set_print_context(), and set_print_options().

Referenced by dflt(), index_dimensions(), no_index_dimensions(), and set_print_options().

7.1.3.720 operator<<() [7/16]

```
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const ex & e)
```

References get_print_context(), operator<<(), GiNaC::ex::print(), and GiNaC::print_dflt::print_dflt().

7.1.3.721 operator>>() [3/3]

```
std::istream & GiNaC::operator>> (
            std::istream & is,
            ex & e)
```

References operator>>().

7.1.3.722 dflt()

```
std::ostream & GiNaC::dflt (
            std::ostream & os)
```

References dflt(), GiNaC::print_dflt::print_dflt(), set_print_context(), and set_print_options().

Referenced by dflt().

7.1.3.723 latex()

```
std::ostream & GiNaC::latex (
            std::ostream & os)
```

References latex(), GiNaC::print_latex::print_latex(), and set_print_context().

Referenced by latex().

7.1.3.724 python()

```
std::ostream & GiNaC::python (
            std::ostream & os)
```

References GiNaC::print_python::print_python(), python(), and set_print_context().

Referenced by python().

7.1.3.725 python_repr()

```
std::ostream & GiNaC::python_repr (
            std::ostream & os)
```

References GiNaC::print_python_repr::print_python_repr(), python_repr(), and set_print_context().

Referenced by python_repr().

7.1.3.726 tree()

```
std::ostream & GiNaC::tree (
            std::ostream & os)
```

References GiNaC::print_tree::print_tree(), set_print_context(), and tree().

Referenced by GiNaC::class_info< OPT >::dump_hierarchy(), and tree().

7.1.3.727 csrc()

```
std::ostream & GiNaC::csrc (
            std::ostream & os)
```

References csrc(), GiNaC::print_csrc_double::print_csrc_double(), and set_print_context().

Referenced by csrc().

7.1.3.728 csrc_float()

```
std::ostream & GiNaC::csrc_float (
            std::ostream & os)
```

References csrc_float(), GiNaC::print_csrc_float::print_csrc_float(), and set_print_context().

Referenced by csrc_float().

7.1.3.729 csrc_double()

```
std::ostream & GiNaC::csrc_double (
            std::ostream & os)
```

References csrc_double(), GiNaC::print_csrc_double::print_csrc_double(), and set_print_context().

Referenced by csrc_double().

7.1.3.730 csrc_cl_N()

```
std::ostream & GiNaC::csrc_cl_N (
            std::ostream & os)
```

References csrc_cl_N(), GiNaC::print_csrc_cl_N::print_csrc_cl_N(), and set_print_context().

Referenced by csrc_cl_N().

7.1.3.731 index_dimensions()

```
std::ostream & GiNaC::index_dimensions (
            std::ostream & os)
```

References get_print_options(), index_dimensions(), GiNaC::print_options::print_index_dimensions, and set_print_options().

Referenced by index_dimensions().

7.1.3.732 no_index_dimensions()

```
std::ostream & GiNaC::no_index_dimensions (
            std::ostream & os)
```

References get_print_options(), no_index_dimensions(), GiNaC::print_options::print_index_dimensions, and set_print_options().

Referenced by no_index_dimensions().

7.1.3.733 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [28/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            power ,
            basic ,
            print_func< print_dflt > &::do_print_dflt.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_csrc > &::do_print_csrc.  print_func< print_python > &::do_↩
print_python.  print_func< print_python_repr > &::do_print_python_repr.  print_func< print_csrc_cl_N
> &::do_print_csrc_cl_N )
```

References GiNaC::power::do_print_csrc(), GiNaC::power::do_print_csrc_cl_N(), GiNaC::power::do_print_dflt(), GiNaC::power::do_print_latex(), GiNaC::power::do_print_python(), GiNaC::power::do_print_python_repr(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.734 print_sym_pow()

```
void GiNaC::print_sym_pow (
            const print_context & c,
            const symbol & x,
            int exp)  [static]
```

References exp(), GiNaC::basic::print(), print_sym_pow(), and GiNaC::print_context::s.

Referenced by GiNaC::power::do_print_csrc(), and print_sym_pow().

7.1.3.735 GINAC_BIND_UNARCHIVER() [38/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            power )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.736 GINAC_DECLARE_UNARCHIVER() [40/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            power )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.737 pow() [2/3]

```
ex GiNaC::pow (
              const ex & b,
              const ex & e)  [inline]
```

Symbolic exponentiation.

Returns a power-object as a new expression.

Parameters

b	the basis expression
e	the exponent expression

References dynallocate(), and pow().

7.1.3.738 pow() [3/3]

```
template<typename T1, typename T2>
ex GiNaC::pow (
              const T1 & b,
              const T2 & e)  [inline]
```

References dynallocate(), GiNaC::ex::ex(), and pow().

7.1.3.739 sqrt() [2/2]

```
ex GiNaC::sqrt (
              const ex & a)  [inline]
```

Square root expression.

Returns a power-object with exponent 1/2.

References _ex1_2, GiNaC::power::power(), and sqrt().

7.1.3.740 is_a() [3/3]

```
template<class T>
bool GiNaC::is_a (
              const print_context & obj)  [inline]
```

Check if obj is a T, including base classes.

References is_a().

7.1.3.741 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [29/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            pseries ,
            basic ,
            print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_tree > &::do_print_tree.  print_func< print_python > &::do_↩
print_python.  print_func< print_python_repr > &::do_print_python_repr )
```

References GiNaC::pseries::do_print(), GiNaC::pseries::do_print_latex(), GiNaC::pseries::do_print_python(), GiNaC::pseries::do_print_python_repr(), GiNaC::pseries::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.742 GINAC_BIND_UNARCHIVER() [39/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            pseries )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.743 GINAC_DECLARE_UNARCHIVER() [41/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            pseries )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.744 series_to_poly()

```
ex GiNaC::series_to_poly (
            const ex & e)  [inline]
```

Convert the pseries object embedded in an expression to an ordinary polynomial in the expansion variable.

The result is undefined if the expression does not contain a pseries object at its top level.

Parameters

e	expression

Returns

polynomial expression

See also

is_a<>

pseries::convert_to_poly

References ex_to(), and series_to_poly().

Referenced by Bernoulli_polynomial(), generalised_Bernoulli_number(), GiNaC::modular_form_kernel::is_numeric(), GiNaC::modular_form_kernel::Laurent_series(), and series_to_poly().

7.1.3.745 is_terminating()

```
bool GiNaC::is_terminating (
            const pseries & s)  [inline]
```

References is_terminating(), and GiNaC::pseries::is_terminating().

Referenced by is_terminating().

7.1.3.746 make_return_type_t()

```
template<typename T>
return_type_t GiNaC::make_return_type_t (
            const unsigned rl = 0)  [inline]
```

References make_return_type_t(), GiNaC::return_type_t::rl, and GiNaC::return_type_t::tinfo.

Referenced by make_return_type_t(), GiNaC::add::return_type_tinfo(), GiNaC::clifford::return_type_tinfo(), GiNaC::color::return_type_tinfo(), GiNaC::function::return_type_tinfo(), GiNaC::mul::return_type_tinfo(), GiNaC::ncmul::return_type_ti and GiNaC::function_options::set_return_type().

7.1.3.747 set_print_func() [1/2]

```
template<class Alg, class Ctx, class T, class C>
void GiNaC::set_print_func (
            void  fconst T &, const C &c, unsigned)  [extern]
```

Add or replace a print method.

References set_print_func().

Referenced by set_print_func(), and set_print_func().

7.1.3.748 set_print_func() [2/2]

```
template<class Alg, class Ctx, class T, class C>
void GiNaC::set_print_func (
            void(T::* f )(const C &, unsigned))  [extern]
```

Add or replace a print method.

References set_print_func().

7.1.3.749 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [30/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            relational ,
            basic ,
            print_func< print_context > &::do_print.  print_func< print_tree > &::do_print↩
_tree.  print_func< print_python_repr > &::do_print_python_repr )
```

References GiNaC::relational::do_print(), GiNaC::relational::do_print_python_repr(), GiNaC::basic::do_print_tree(), and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.750 GINAC_BIND_UNARCHIVER() [40/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            relational )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.751 print_operator()

```
void GiNaC::print_operator (
            const print_context & c,
            relational::operators o)  [static]
```

References GiNaC::relational::equal, GiNaC::relational::greater, GiNaC::relational::greater_or_equal, GiNaC::relational::less, GiNaC::relational::less_or_equal, GiNaC::relational::not_equal, print_operator(), and GiNaC::print_context::s.

Referenced by GiNaC::relational::do_print(), GiNaC::relational::do_print_python_repr(), and print_operator().

7.1.3.752 GINAC_DECLARE_UNARCHIVER() [42/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            relational )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.753 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [31/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            symbol ,
            basic ,
            print_func< print_context > &::do_print.  print_func< print_latex > &::do_↩
print_latex.  print_func< print_tree > &::do_print_tree.  print_func< print_python_repr >
&::do_print_python_repr )
```

References GiNaC::symbol::do_print(), GiNaC::symbol::do_print_latex(), GiNaC::symbol::do_print_python_repr(), GiNaC::symbol::do_print_tree(), GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, and GINAC_IMPLEMENT_REGIST

7.1.3.754 get_default_TeX_name()

```
const std::string & GiNaC::get_default_TeX_name (
            const std::string & name)  [static]
```

Return default TeX name for symbol.

This recognizes some greek letters.

References get_default_TeX_name().

Referenced by GiNaC::symbol::do_print_latex(), get_default_TeX_name(), and GiNaC::symbol::get_TeX_name().

7.1.3.755 GINAC_BIND_UNARCHIVER() [41/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            symbol )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.756 GINAC_BIND_UNARCHIVER() [42/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            realsymbol )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.757 GINAC_BIND_UNARCHIVER() [43/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            possymbol )
```

References GINAC_BIND_UNARCHIVER, and GiNaC::symbol::next_serial.

7.1.3.758 GINAC_DECLARE_UNARCHIVER() [43/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            symbol )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.759 GINAC_DECLARE_UNARCHIVER() [44/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            realsymbol )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.760 GINAC_DECLARE_UNARCHIVER() [45/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            possymbol )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.761 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [32/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            symmetry ,
            basic ,
            print_func< print_context > &::do_print.  print_func< print_tree > &::do_print↩
_tree )
```

References GiNaC::symmetry::do_print(), GiNaC::symmetry::do_print_tree(), GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.762 GINAC_BIND_UNARCHIVER() [44/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            symmetry )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.763 index0()

```
const symmetry & GiNaC::index0 ()  [static]
```

References dynallocate(), ex_to(), and index0().

Referenced by antisymmetric2(), antisymmetric3(), antisymmetric4(), index0(), symmetric2(), symmetric3(), and symmetric4().

7.1.3.764 index1()

```
const symmetry & GiNaC::index1 ()  [static]
```

References dynallocate(), ex_to(), and index1().

Referenced by antisymmetric2(), antisymmetric3(), antisymmetric4(), index1(), symmetric2(), symmetric3(), and symmetric4().

7.1.3.765 index2()

```
const symmetry & GiNaC::index2 ()  [static]
```

References dynallocate(), ex_to(), and index2().

Referenced by antisymmetric3(), antisymmetric4(), index2(), symmetric3(), and symmetric4().

7.1.3.766 index3()

```
const symmetry & GiNaC::index3 ()  [static]
```

References dynallocate(), ex_to(), and index3().

Referenced by antisymmetric4(), index3(), and symmetric4().

7.1.3.767 not_symmetric()

```
const symmetry & GiNaC::not_symmetric ()
```

References dynallocate(), ex_to(), and not_symmetric().

Referenced by GiNaC::clifford::clifford(), GiNaC::clifford::clifford(), clifford_unit(), GiNaC::color::color(), GiNaC::color::color(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GiNaC::indexed::indexed(), GiNaC::indexed::indexed(), GiNaC::indexed::indexed(), GiNaC::indexed::indexed(), GiNaC::indexed::indexed(), GiNaC::indexed::indexed(), not_symmetric(), and GiNaC::indexed::read_archive().

7.1.3.768 symmetric2()

```
const symmetry & GiNaC::symmetric2 ()
```

References dynallocate(), ex_to(), index0(), index1(), GiNaC::symmetry::symmetric, and symmetric2().

Referenced by clifford_unit(), delta_tensor(), dirac_gamma(), dirac_slash(), GiNaC::clifford::get_metric(), lorentz_g(), metric_tensor(), and symmetric2().

7.1.3.769 symmetric3()

```
const symmetry & GiNaC::symmetric3 ()
```

References dynallocate(), ex_to(), index0(), index1(), index2(), GiNaC::symmetry::symmetric, and symmetric3().

Referenced by color_d(), and symmetric3().

7.1.3.770 symmetric4()

```
const symmetry & GiNaC::symmetric4 ()
```

References dynallocate(), ex_to(), index0(), index1(), index2(), index3(), GiNaC::symmetry::symmetric, and symmetric4().

Referenced by symmetric4().

7.1.3.771 antisymmetric2()

```
const symmetry & GiNaC::antisymmetric2 ()
```

References GiNaC::symmetry::antisymmetric, antisymmetric2(), dynallocate(), ex_to(), index0(), and index1().

Referenced by antisymmetric2(), epsilon_tensor(), and spinor_metric().

7.1.3.772 antisymmetric3()

```
const symmetry & GiNaC::antisymmetric3 ()
```

References GiNaC::symmetry::antisymmetric, antisymmetric3(), dynallocate(), ex_to(), index0(), index1(), and index2().

Referenced by antisymmetric3(), color_f(), and epsilon_tensor().

7.1.3.773 antisymmetric4()

```
const symmetry & GiNaC::antisymmetric4 ()
```

References GiNaC::symmetry::antisymmetric, antisymmetric4(), dynallocate(), ex_to(), index0(), index1(), index2(), and index3().

Referenced by antisymmetric4(), and lorentz_eps().

7.1.3.774 canonicalize()

```
int GiNaC::canonicalize (
            exvector::iterator v,
            const symmetry & symm)
```

Canonicalize the order of elements of an expression vector, according to the symmetry properties defined in a symmetry tree.

Parameters

v	Start of expression vector
symm	Root node of symmetry tree

Returns

the overall sign introduced by the reordering (+1, -1 or 0) or numeric_limits<int>::max() if nothing changed

Referenced by GiNaC::function::eval(), and GiNaC::indexed::eval().

7.1.3.775 symm()

```
ex GiNaC::symm (
            const ex & e,
            exvector::const_iterator first,
            exvector::const_iterator last,
            bool asymmetric)  [static]
```

References GiNaC::container< C >::append(), class_info< OPT >::first, dynallocate(), factorial(), GiNaC::subs_options::no_index_re GiNaC::subs_options::no_pattern, GiNaC::numeric::numeric(), GiNaC::container< C >::op(), permutation_sign(), GiNaC::ex::subs(), and symm().

Referenced by antisymmetrize(), GiNaC::ex::antisymmetrize(), GiNaC::symmetry::canonicalize, GiNaC::symmetry::sy_swap, symm(), GiNaC::ex::symmetrize(), and symmetrize().

7.1.3.776 symmetrize() [3/4]

```
ex GiNaC::symmetrize (
            const ex & e,
            exvector::const_iterator first,
            exvector::const_iterator last)
```

Symmetrize expression over a set of objects (symbols, indices).

References class_info< OPT >::first, symm(), and symmetrize().

7.1.3.777 antisymmetrize() [3/4]

```
ex GiNaC::antisymmetrize (
            const ex & e,
            exvector::const_iterator first,
            exvector::const_iterator last)
```

Antisymmetrize expression over a set of objects (symbols, indices).

References antisymmetrize(), class_info< OPT >::first, and symm().

7.1.3.778 symmetrize_cyclic() [3/4]

```
ex GiNaC::symmetrize_cyclic (
            const ex & e,
            exvector::const_iterator first,
            exvector::const_iterator last)
```

Symmetrize expression by cyclic permutation over a set of objects (symbols, indices).

References GiNaC::container< C >::append(), class_info< OPT >::first, GiNaC::subs_options::no_index_renaming, GiNaC::subs_options::no_pattern, GiNaC::container< C >::op(), GiNaC::container< C >::remove_first(), GiNaC::ex::subs(), and symmetrize_cyclic().

7.1.3.779 GINAC_DECLARE_UNARCHIVER() [46/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            symmetry )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.780 sy_none() [1/4]

```
symmetry GiNaC::sy_none ()  [inline]
```

References sy_none(), and GiNaC::symmetry::symmetry().

Referenced by sy_none(), sy_none(), sy_none(), and sy_none().

7.1.3.781 sy_none() [2/4]

```
symmetry GiNaC::sy_none (
            const symmetry & c1,
            const symmetry & c2)  [inline]
```

References GiNaC::symmetry::none, sy_none(), and GiNaC::symmetry::symmetry().

7.1.3.782 sy_none() [3/4]

```
symmetry GiNaC::sy_none (
            const symmetry & c1,
            const symmetry & c2,
            const symmetry & c3)  [inline]
```

References GiNaC::symmetry::add(), GiNaC::symmetry::none, sy_none(), and GiNaC::symmetry::symmetry().

7.1.3.783 sy_none() [4/4]

```
symmetry GiNaC::sy_none (
              const symmetry & c1,
              const symmetry & c2,
              const symmetry & c3,
              const symmetry & c4)  [inline]
```

References GiNaC::symmetry::add(), GiNaC::symmetry::none, sy_none(), and GiNaC::symmetry::symmetry().

7.1.3.784 sy_symm() [1/4]

```
symmetry GiNaC::sy_symm ()  [inline]
```

References GiNaC::symmetry::set_type(), sy_symm(), and GiNaC::symmetry::symmetric.

Referenced by GiNaC::indexed::read_archive(), REGISTER_FUNCTION(), REGISTER_FUNCTION(), sy_symm(), sy_symm(), sy_symm(), and sy_symm().

7.1.3.785 sy_symm() [2/4]

```
symmetry GiNaC::sy_symm (
              const symmetry & c1,
              const symmetry & c2)  [inline]
```

References sy_symm(), GiNaC::symmetry::symmetric, and GiNaC::symmetry::symmetry().

7.1.3.786 sy_symm() [3/4]

```
symmetry GiNaC::sy_symm (
              const symmetry & c1,
              const symmetry & c2,
              const symmetry & c3)  [inline]
```

References GiNaC::symmetry::add(), sy_symm(), GiNaC::symmetry::symmetric, and GiNaC::symmetry::symmetry().

7.1.3.787 sy_symm() [4/4]

```
symmetry GiNaC::sy_symm (
              const symmetry & c1,
              const symmetry & c2,
              const symmetry & c3,
              const symmetry & c4)  [inline]
```

References GiNaC::symmetry::add(), sy_symm(), GiNaC::symmetry::symmetric, and GiNaC::symmetry::symmetry().

7.1.3.788 sy_anti() [1/4]

```
symmetry GiNaC::sy_anti ()  [inline]
```

References GiNaC::symmetry::antisymmetric, GiNaC::symmetry::set_type(), and sy_anti().

Referenced by GiNaC::indexed::read_archive(), sy_anti(), sy_anti(), sy_anti(), and sy_anti().

7.1.3.789 sy_anti() [2/4]

```
symmetry GiNaC::sy_anti (
            const symmetry & c1,
            const symmetry & c2)  [inline]
```

References GiNaC::symmetry::antisymmetric, sy_anti(), and GiNaC::symmetry::symmetry().

7.1.3.790 sy_anti() [3/4]

```
symmetry GiNaC::sy_anti (
            const symmetry & c1,
            const symmetry & c2,
            const symmetry & c3)  [inline]
```

References GiNaC::symmetry::add(), GiNaC::symmetry::antisymmetric, sy_anti(), and GiNaC::symmetry::symmetry().

7.1.3.791 sy_anti() [4/4]

```
symmetry GiNaC::sy_anti (
            const symmetry & c1,
            const symmetry & c2,
            const symmetry & c3,
            const symmetry & c4)  [inline]
```

References GiNaC::symmetry::add(), GiNaC::symmetry::antisymmetric, sy_anti(), and GiNaC::symmetry::symmetry().

7.1.3.792 sy_cycl() [1/4]

```
symmetry GiNaC::sy_cycl ()  [inline]
```

References GiNaC::symmetry::cyclic, GiNaC::symmetry::set_type(), and sy_cycl().

Referenced by sy_cycl(), sy_cycl(), sy_cycl(), and sy_cycl().

7.1.3.793 sy_cycl() [2/4]

```
symmetry GiNaC::sy_cycl (
            const symmetry & c1,
            const symmetry & c2)  [inline]
```

References GiNaC::symmetry::cyclic, sy_cycl(), and GiNaC::symmetry::symmetry().

7.1.3.794 sy_cycl() [3/4]

```
symmetry GiNaC::sy_cycl (
             const symmetry & c1,
             const symmetry & c2,
             const symmetry & c3)  [inline]
```

References GiNaC::symmetry::add(), GiNaC::symmetry::cyclic, sy_cycl(), and GiNaC::symmetry::symmetry().

7.1.3.795 sy_cycl() [4/4]

```
symmetry GiNaC::sy_cycl (
             const symmetry & c1,
             const symmetry & c2,
             const symmetry & c3,
             const symmetry & c4)  [inline]
```

References GiNaC::symmetry::add(), GiNaC::symmetry::cyclic, sy_cycl(), and GiNaC::symmetry::symmetry().

7.1.3.796 symmetrize() [4/4]

```
ex GiNaC::symmetrize (
             const ex & e,
             const exvector & v)  [inline]
```

Symmetrize expression over a set of objects (symbols, indices).

References GiNaC::ex::begin(), and symmetrize().

7.1.3.797 antisymmetrize() [4/4]

```
ex GiNaC::antisymmetrize (
             const ex & e,
             const exvector & v)  [inline]
```

Antisymmetrize expression over a set of objects (symbols, indices).

References antisymmetrize(), and GiNaC::ex::begin().

7.1.3.798 symmetrize_cyclic() [4/4]

```
ex GiNaC::symmetrize_cyclic (
             const ex & e,
             const exvector & v)  [inline]
```

Symmetrize expression by cyclic permutation over a set of objects (symbols, indices).

References GiNaC::ex::begin(), symmetrize(), and symmetrize_cyclic().

7.1.3.799 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [33/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
             tensdelta ,
             tensor ,
             print_func< print_dflt > &::do_print.  print_func< print_latex > &::do_print_↩
latex )
```

References GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.800 print_func< print_dflt >() [3/3]

```
GiNaC::print_func< print_dflt > (
             &tensmetric::do_print ) &
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT

7.1.3.801 GINAC_BIND_UNARCHIVER() [45/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             minkmetric )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.802 GINAC_BIND_UNARCHIVER() [46/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             tensepsilon )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.803 GINAC_BIND_UNARCHIVER() [47/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             tensdelta )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.804 GINAC_BIND_UNARCHIVER() [48/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
             tensmetric )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.805 GINAC_BIND_UNARCHIVER() [49/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
              spinmetric )
```

References DEFAULT_COMPARE, and GINAC_BIND_UNARCHIVER.

7.1.3.806 delta_tensor()

```
ex GiNaC::delta_tensor (
              const ex & i1,
              const ex & i2)
```

Create a delta tensor with specified indices.

The indices must be of class idx or a subclass. The delta tensor is always symmetric and its trace is the dimension of the index space.

Parameters

i1	First index
i2	Second index

Returns

newly constructed delta tensor

References delta_tensor(), dynallocate(), GiNaC::indexed::indexed(), is_a(), and symmetric2().

Referenced by color_trace(), GiNaC::spinmetric::contract_with(), GiNaC::su3d::contract_with(), GiNaC::su3f::contract_with(), GiNaC::tensepsilon::contract_with(), delta_tensor(), GiNaC::tensmetric::eval_indexed(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.807 metric_tensor()

```
ex GiNaC::metric_tensor (
              const ex & i1,
              const ex & i2)
```

Create a symmetric metric tensor with specified indices.

The indices must be of class varidx or a subclass. A metric tensor with one covariant and one contravariant index is equivalent to the delta tensor.

Parameters

i1	First index
i2	Second index

Returns

newly constructed metric tensor

References dynallocate(), GiNaC::indexed::indexed(), is_a(), metric_tensor(), and symmetric2().

Referenced by GiNaC::tensepsilon::contract_with(), GINAC_DECLARE_UNARCHIVER(), and metric_tensor().

7.1.3.808 lorentz_g()

```
ex GiNaC::lorentz_g (
             const ex & i1,
             const ex & i2,
             bool pos_sig = false)
```

Create a Minkowski metric tensor with specified indices.

The indices must be of class varidx or a subclass. The Lorentz metric is a symmetric tensor with a matrix representation of diag(1,-1,-1,...) (negative signature, the default) or diag(-1,1,1,...) (positive signature).

Parameters

i1	First index
i2	Second index
pos_sig	Whether the signature is positive

Returns

newly constructed Lorentz metric tensor

References dynallocate(), GiNaC::indexed::indexed(), is_a(), lorentz_g(), and symmetric2().

Referenced by GiNaC::diracgamma::contract_with(), GiNaC::tensepsilon::contract_with(), dirac_trace(), GINAC_DECLARE_UNARCH lorentz_g(), and trace_string().

7.1.3.809 spinor_metric()

```
ex GiNaC::spinor_metric (
             const ex & i1,
             const ex & i2)
```

Create a spinor metric tensor with specified indices.

The indices must be of class spinidx or a subclass and have a dimension of 2. The spinor metric is an antisymmetric tensor with a matrix representation of [[[[0, 1]], [[-1, 0]]]].

Parameters

i1	First index
i2	Second index

Returns

newly constructed spinor metric tensor

References antisymmetric2(), dynallocate(), ex_to(), GiNaC::indexed::indexed(), is_a(), and spinor_metric().

Referenced by GINAC_DECLARE_UNARCHIVER(), and spinor_metric().

7.1.3.810 epsilon_tensor() [1/2]

```
ex GiNaC::epsilon_tensor (
              const ex & i1,
              const ex & i2)
```

Create an epsilon tensor in a Euclidean space with two indices.

The indices must be of class idx or a subclass, and have a dimension of 2.

Parameters

i1	First index
i2	Second index

Returns

newly constructed epsilon tensor

References _ex2, antisymmetric2(), dynallocate(), epsilon_tensor(), ex_to(), GiNaC::indexed::indexed(), is_a(), GiNaC::ex::is_equal(), and GiNaC::ex::op().

Referenced by epsilon_tensor(), epsilon_tensor(), and GINAC_DECLARE_UNARCHIVER().

7.1.3.811 epsilon_tensor() [2/2]

```
ex GiNaC::epsilon_tensor (
              const ex & i1,
              const ex & i2,
              const ex & i3)
```

Create an epsilon tensor in a Euclidean space with three indices.

The indices must be of class idx or a subclass, and have a dimension of 3.

Parameters

i1	First index
i2	Second index
i3	Third index

Returns

newly constructed epsilon tensor

References _ex3, antisymmetric3(), dynallocate(), epsilon_tensor(), ex_to(), GiNaC::indexed::indexed(), is_a(), GiNaC::ex::is_equal(), and GiNaC::ex::op().

7.1.3.812 lorentz_eps()

```
ex GiNaC::lorentz_eps (
              const ex & i1,
              const ex & i2,
              const ex & i3,
              const ex & i4,
              bool pos_sig = false)
```

Create an epsilon tensor in a Minkowski space with four indices.

The indices must be of class varidx or a subclass, and have a dimension of 4.

Parameters

i1	First index
i2	Second index
i3	Third index
i4	Fourth index
pos_sig	Whether the signature of the metric is positive

Returns

newly constructed epsilon tensor

References _ex4, antisymmetric4(), dynallocate(), ex_to(), GiNaC::indexed::indexed(), is_a(), GiNaC::ex::is_equal(), lorentz_eps(), and GiNaC::ex::op().

Referenced by dirac_trace(), GINAC_DECLARE_UNARCHIVER(), and lorentz_eps().

7.1.3.813 GINAC_DECLARE_UNARCHIVER() [47/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
              tensdelta )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.814 GINAC_DECLARE_UNARCHIVER() [48/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
              tensmetric )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.815 GINAC_DECLARE_UNARCHIVER() [49/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
              minkmetric )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.816 GINAC_DECLARE_UNARCHIVER() [50/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            spinmetric )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.817 GINAC_DECLARE_UNARCHIVER() [51/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            tensepsilon )
```

References delta_tensor(), epsilon_tensor(), GINAC_DECLARE_UNARCHIVER(), lorentz_eps(), lorentz_g(), metric_tensor(), and spinor_metric().

7.1.3.818 log2()

```
unsigned GiNaC::log2 (
            unsigned n)
```

Integer binary logarithm.

References log2().

Referenced by log2(), and GiNaC::remember_table::remember_table().

7.1.3.819 multinomial_coefficient()

```
const numeric GiNaC::multinomial_coefficient (
            const std::vector< unsigned > & p)
```

Compute the multinomial coefficient n!

/(p1!∗p2!∗...∗pk!) where n = p1+p2+...+pk, i.e. p is a partition of n.

References GiNaC::numeric::add(), GiNaC::numeric::div(), factorial(), GiNaC::numeric::mul(), multinomial_coefficient(), and GiNaC::numeric::numeric().

Referenced by GiNaC::power::expand_add(), and multinomial_coefficient().

7.1.3.820 rotate_left()

```
unsigned GiNaC::rotate_left (
            unsigned n)  [inline]
```

Rotate bits of unsigned value by one bit to the left.

This can be necessary if the user wants to define its own hashes.

References rotate_left().

Referenced by GiNaC::basic::calchash(), GiNaC::expairseq::calchash(), GiNaC::function::calchash(), GiNaC::Gt::calchash(), GiNaC::idx::calchash(), GiNaC::relational::calchash(), GiNaC::symmetry::calchash(), and rotate_left().

7.1.3.821 compare_pointers()

```
template<class T>
int GiNaC::compare_pointers (
             const T * a,
             const T * b)  [inline]
```

Compare two pointers (just to establish some sort of canonical order).

Returns

-1, 0, or 1

References compare_pointers().

Referenced by compare_pointers(), and GiNaC::basic::compare_same_type().

7.1.3.822 golden_ratio_hash()

```
unsigned GiNaC::golden_ratio_hash (
             uintptr_t n)  [inline]
```

Truncated multiplication with golden ratio, for computing hash values.

References golden_ratio_hash().

Referenced by GiNaC::constant::calchash(), GiNaC::function::calchash(), GiNaC::numeric::calchash(), GiNaC::symbol::calchash(), GiNaC::wildcard::calchash(), and golden_ratio_hash().

7.1.3.823 permutation_sign() [1/2]

```
template<class It>
int GiNaC::permutation_sign (
             It first,
             It last)
```

References class_info< OPT >::first, permutation_sign(), swap(), and std::swap().

Referenced by GiNaC::symmetry::canonicalize, GiNaC::matrix::determinant(), dirac_trace(), GiNaC::tensepsilon::eval_indexed(), permutation_sign(), permutation_sign(), and symm().

7.1.3.824 permutation_sign() [2/2]

```
template<class It, class Cmp, class Swap>
int GiNaC::permutation_sign (
             It first,
             It last,
             Cmp comp,
             Swap swapit)
```

References class_info< OPT >::first, and permutation_sign().

7.1.3.825 shaker_sort()

```
template<class It, class Cmp, class Swap>
void GiNaC::shaker_sort (
             It first,
             It last,
             Cmp comp,
             Swap swapit)
```

References class_info< OPT >::first, and shaker_sort().

Referenced by GiNaC::symmetry::canonicalize, find_free_and_dummy(), rename_dummy_indices(), and shaker_sort().

7.1.3.826 cyclic_permutation()

```
template<class It, class Swap>
void GiNaC::cyclic_permutation (
             It first,
             It last,
             It new_first,
             Swap swapit)
```

References class_info< OPT >::first, and cyclic_permutation().

Referenced by GiNaC::symmetry::canonicalize, and cyclic_permutation().

7.1.3.827 format_index_value() [1/2]

```
template<typename T>
std::enable_if< has_distance< T >::value, typenamestd::iterator_traits< T >::difference_type
>::type GiNaC::format_index_value (
             const T & a,
             const T & b)
```

For printing a multi-index: If the templates are used, where T is an iterator, printing the address where the iterator points to is not meaningful.

However, we may print the difference to the starting point.

References format_index_value().

Referenced by format_index_value(), operator<<(), operator<<(), operator<<(), operator<<(), operator<<(), operator<<(), operator<<(), operator<<(), and operator<<().

7.1.3.828 format_index_value() [2/2]

```
template<typename T>
std::enable_if<!has_distance< T >::value, T >::type GiNaC::format_index_value (
             const T & a,
             const T & b)
```

For all other cases we simply print the value.

References format_index_value().

7.1.3.829 operator<<() [8/16]

```
template<class T>
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const basic_multi_iterator< T > & v)  [inline]
```

Output operator.

A multi_iterator prints out as basic_multi_iterator($n_0, n_1, ...$).

References GiNaC::basic_multi_iterator< T >::B, format_index_value(), and GiNaC::basic_multi_iterator< T >::size().

7.1.3.830 operator<<() [9/16]

```
template<class T>
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const multi_iterator_ordered< T > & v)  [inline]
```

Output operator.

A multi_iterator_ordered prints out as multi_iterator_ordered($n_0, n_1, ...$).

References GiNaC::basic_multi_iterator< T >::B, format_index_value(), and GiNaC::basic_multi_iterator< T >::size().

7.1.3.831 operator<<() [10/16]

```
template<class T>
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const multi_iterator_ordered_eq< T > & v)  [inline]
```

Output operator.

A multi_iterator_ordered_eq prints out as multi_iterator_ordered_eq($n_0, n_1, ...$).

References GiNaC::basic_multi_iterator< T >::B, format_index_value(), and GiNaC::basic_multi_iterator< T >::size().

7.1.3.832 operator<<() [11/16]

```
template<class T>
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const multi_iterator_ordered_eq_indv< T > & v)  [inline]
```

Output operator.

A multi_iterator_ordered_eq_indv prints out as multi_iterator_ordered_eq_indv($n_0, n_1, ...$).

References GiNaC::basic_multi_iterator< T >::B, format_index_value(), and GiNaC::basic_multi_iterator< T >::size().

7.1.3.833 operator<<() [12/16]

```
template<class T>
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const multi_iterator_counter< T > & v)  [inline]
```

Output operator.

A multi_iterator_counter prints out as multi_iterator_counter($n_0, n_1, ...$).

References GiNaC::basic_multi_iterator< T >::B, format_index_value(), and GiNaC::basic_multi_iterator< T >::size().

7.1.3.834 operator<<() [13/16]

```
template<class T>
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const multi_iterator_counter_indv< T > & v)  [inline]
```

Output operator.

A multi_iterator_counter_indv prints out as multi_iterator_counter_indv($n_0, n_1, ...$).

References GiNaC::basic_multi_iterator< T >::B, format_index_value(), and GiNaC::basic_multi_iterator< T >::size().

7.1.3.835 operator<<() [14/16]

```
template<class T>
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const multi_iterator_permutation< T > & v)  [inline]
```

Output operator.

A multi_iterator_permutation prints out as multi_iterator_permutation($n_0, n_1, ...$).

References GiNaC::basic_multi_iterator< T >::B, format_index_value(), and GiNaC::basic_multi_iterator< T >::size().

7.1.3.836 operator<<() [15/16]

```
template<class T>
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const multi_iterator_shuffle< T > & v)  [inline]
```

Output operator.

A multi_iterator_shuffle prints out as multi_iterator_shuffle($n_0, n_1, ...$).

References GiNaC::basic_multi_iterator< T >::B, format_index_value(), and GiNaC::basic_multi_iterator< T >::size().

7.1.3.837 operator<<() [16/16]

```
template<class T>
std::ostream & GiNaC::operator<< (
            std::ostream & os,
            const multi_iterator_shuffle_prime< T > & v)  [inline]
```

Output operator.

A multi_iterator_shuffle_prime prints out as multi_iterator_shuffle_prime($n_0, n_1, ...$).

References GiNaC::basic_multi_iterator< T >::B, format_index_value(), and GiNaC::basic_multi_iterator< T >::size().

7.1.3.838 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT() [34/34]

```
GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (
            wildcard ,
            basic ,
            print_func< print_context > &::do_print.  print_func< print_tree > &::do_print↩
_tree.  print_func< print_python_repr > &::do_print_python_repr )
```

References GiNaC::wildcard::do_print(), GiNaC::wildcard::do_print_python_repr(), GiNaC::wildcard::do_print_tree(), GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, and GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

7.1.3.839 GINAC_BIND_UNARCHIVER() [50/50]

```
GiNaC::GINAC_BIND_UNARCHIVER (
            wildcard )
```

References GINAC_BIND_UNARCHIVER.

7.1.3.840 haswild()

```
bool GiNaC::haswild (
            const ex & x)
```

Check whether x has a wildcard anywhere as a subexpression.

References haswild(), is_a(), GiNaC::ex::nops(), and GiNaC::ex::op().

Referenced by GiNaC::integral::eval(), and haswild().

7.1.3.841 GINAC_DECLARE_UNARCHIVER() [52/52]

```
GiNaC::GINAC_DECLARE_UNARCHIVER (
            wildcard )
```

References GINAC_DECLARE_UNARCHIVER().

7.1.3.842 wild()

```
ex GiNaC::wild (
            unsigned label = 0)  [inline]
```

Create a wildcard object with the specified label.

References wild(), and GiNaC::wildcard::wildcard().

Referenced by GiNaC::spmapkey::spmapkey(), and wild().

7.1.4 Variable Documentation

7.1.4.1 unarch_table_instance

```
unarchive_table_t GiNaC::unarch_table_instance  [static]
```

7.1.4.2 map_evalm

```
GiNaC::evalm_map_function GiNaC::map_evalm
```

Referenced by GiNaC::basic::evalm().

7.1.4.3 map_eval_integ

```
GiNaC::eval_integ_map_function GiNaC::map_eval_integ
```

Referenced by GiNaC::basic::eval_integ().

7.1.4.4 class_info< OPT >::first

```
template<class OPT>
class_info<OPT>* GiNaC::class_info< OPT >::first = nullptr
```

Referenced by antisymmetrize(), GiNaC::symmetry::canonicalize, GiNaC::expairseq::construct_from_expairseq_ex(), cyclic_permutation(), GiNaC::mul::do_print(), GiNaC::clifford::eval_ncmul(), GiNaC::Gt_detail::pathintegral_term::integrate(), permutation_sign(), permutation_sign(), GiNaC::add::print_add(), GiNaC::indexed::printindices(), GiNaC::Gt::qExpand(), shaker_sort(), GiNaC::indexed::simplify_indexed, symm(), symmetrize(), symmetrize_cyclic(), and GiNaC::Gt::zisToFundamental().

7.1.4.5 class_info< OPT >::parents_identified

```
template<class OPT>
bool GiNaC::class_info< OPT >::parents_identified = false
```

7.1.4.6 tensor

```
GiNaC::tensor
```

7.1.4.7 Pi

```
const constant GiNaC::Pi (
             "Pi" ,
             PiEvalf ,
             "\\pi" ,
             domain::positive )
```

Pi.

(3.14159...) Diverts straight into CLN for evalf().

Referenced by acos_eval(), acosh_eval(), asin_eval(), atan2_eval(), atan_eval(), atan_series(), atanh_series(), GiNaC::Eisenstein_h_kernel::coefficient_a0(), GiNaC::Eisenstein_h_kernel::coefficient_an(), cos_eval(), cosh_eval(), EllipticE_eval(), EllipticE_evalf(), EllipticE_series(), EllipticK_eval(), EllipticK_evalf(), EllipticK_series(), eta_eval(), eta_evalf(), exp_eval(), GiNaC::Kronecker_dtau_kernel::get_numerical_value(), GiNaC::Kronecker_dz_kernel::get_numerical_value(), H_evalf(), Li2_eval(), Li2_series(), Li_eval(), log_eval(), log_series(), GiNaC::Gt::qExpand(), GiNaC::constant::read_archive(), GiNaC::Kronecker_dtau_kernel::series_coeff_impl(), GiNaC::Kronecker_dz_kernel::series_coeff_impl(), sin_eval(), sinh_eval(), tan_eval(), tan_series(), tanh_eval(), tanh_series(), GiNaC::Gt::tauToFundamental(), tgamma_eval(), zeta1_eval(), and GiNaC::Gt::zisToFundamental().

7.1.4.8 Euler

```
const constant GiNaC::Euler (
             "Euler" ,
             EulerEvalf ,
             "\\gamma_E" ,
             domain::positive )
```

Euler's constant.

(0.57721...) Sometimes called Euler-Mascheroni constant. Diverts straight into CLN for evalf().

Referenced by psi1_eval(), and GiNaC::constant::read_archive().

7.1.4.9 Catalan

```
const constant GiNaC::Catalan (
             "Catalan" ,
             CatalanEvalf ,
             "G" ,
             domain::positive )
```

Catalan's constant.

(0.91597...) Diverts straight into CLN for evalf().

Referenced by Li2_eval(), Li_eval(), and GiNaC::constant::read_archive().

7.1.4.10 crctab

```
unsigned const GiNaC::crctab[256]  [static]
```

Referenced by crc32().

7.1.4.11 library_initializer

```
library_init GiNaC::library_initializer [static]
```

For construction of flyweights, etc.

7.1.4.12 _num0_bp

```
const basic * GiNaC::_num0_bp
```

Referenced by GiNaC::ex::ex(), and GiNaC::library_init::library_init().

7.1.4.13 idx

```
GiNaC::idx
```

7.1.4.14 force_include_tgamma

```
unsigned GiNaC::force_include_tgamma = tgamma_SERIAL::serial
```

7.1.4.15 force_include_zeta1

```
unsigned GiNaC::force_include_zeta1 = zeta1_SERIAL::serial
```

7.1.4.16 GINAC_BIND_UNARCHIVER

```
template<>
GINAC_IMPLEMENT_REGISTERED_CLASS_OPT_T(lst, basic, print_func< print_context >(&lst::do_print).
print_func< print_tree >(&lst::do_print_tree)) template<> bool lst GiNaC::GINAC_BIND_UNARCHIVER(lst)
(
            lst )
```

Specialization of container::info() for lst.

Referenced by GINAC_BIND_UNARCHIVER(), and GINAC_IMPLEMENT_REGIS

7.1.4.17 I

```
const numeric GiNaC::I = numeric(cln::complex(cln::cl_I(0),cln::cl_I(1)))
```

Imaginary unit.

This is not a constant but a numeric since we are natively handing complex numbers anyways, so in each expression containing an I it is automatically eval'ed away anyhow.

Referenced by acosh_eval(), atan_eval(), atan_series(), atanh_series(), GiNaC::Eisenstein_h_kernel::coefficient_a0(), GiNaC::Eisenstein_h_kernel::coefficient_an(), color_h(), GiNaC::su3f::contract_with(), cosh_eval(), csgn_eval(), dirac_trace(), eta_eval(), eta_evalf(), eta_imag_part(), exp_eval(), GiNaC::Kronecker_dtau_kernel::get_numerical_value(), GiNaC::Kronecker_dz_kernel::get_numerical_value(), GINAC_DECLARE_UNARCHIVER(), H_evalf(), GiNaC::numeric::has(), Li2_eval(), Li2_series(), Li_eval(), log_eval(), log_series(), GiNaC::numeric::normal(), GiNaC::Gt::qExpand(), GiNaC::Kronecker_dtau_kernel::series_coeff_impl(), GiNaC::Kronecker_dz_kernel::series_coeff_impl(), sinh_eval(), step_eval(), tanh_eval(), tanh_series(), GiNaC::Gt::tauToFundamental(), GiNaC::numeric::to_polynomial(), GiNaC::numeric::to_rational(), and GiNaC::Gt::zisToFundamental().

7.1.4.18 Digits

```
_numeric_digits GiNaC::Digits
```

Accuracy in decimal digits.

Only object of this type! Can be set using assignment from C++ unsigned ints and evaluated like any built-in type.

Referenced by GiNaC::integration_kernel::get_numerical_value_impl(), GINAC_DECLARE_UNARCHIVER(), iterated_integral_evalf_impl(), GiNaC::numeric::numeric(), print_real_cl_N(), GiNaC::Gt::qExpand(), and zeta1_evalf().

7.1.4.19 next_print_context_id

```
unsigned GiNaC::next_print_context_id = 0
```

Next free ID for print_context types.

7.1.4.20 structure< T, CP >::reg_info

```
template<class T, template< class > class CP>
registered_class_info GiNaC::structure< T, CP >::reg_info = registered_class_info(registered_class_options(st
"basic", typeid(structure<T, CP>)))
```

Referenced by GiNaC::basic::print_dispatch().

7.1.4.21 version_major

```
const int GiNaC::version_major = GINACLIB_MAJOR_VERSION
```

7.1.4.22 version_minor

```
const int GiNaC::version_minor = GINACLIB_MINOR_VERSION
```

7.1.4.23 version_micro

```
const int GiNaC::version_micro = GINACLIB_MICRO_VERSION
```

7.1.4.24 _num_120_p

```
const numeric * GiNaC::_num_120_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.25 _ex_120

```
const ex GiNaC::_ex_120 = ex(*_num_120_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.26 _num_60_p

```
const numeric * GiNaC::_num_60_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.27 _ex_60

```
const ex GiNaC::_ex_60 = ex(*_num_60_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.28 _num_48_p

```
const numeric * GiNaC::_num_48_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.29 _ex_48

```
const ex GiNaC::_ex_48 = ex(*_num_48_p)
```

Referenced by Li2_eval(), Li_eval(), GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.30 _num_30_p

```
const numeric * GiNaC::_num_30_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.31 _ex_30

```
const ex GiNaC::_ex_30 = ex(*_num_30_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.32 _num_25_p

```
const numeric * GiNaC::_num_25_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.33 _ex_25

```
const ex GiNaC::_ex_25 = ex(*_num_25_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.34 _num_24_p

```
const numeric * GiNaC::_num_24_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.35 _ex_24

```
const ex GiNaC::_ex_24 = ex(*_num_24_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.36 _num_20_p

```
const numeric * GiNaC::_num_20_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.37 _ex_20

```
const ex GiNaC::_ex_20 = ex(*_num_20_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.38 _num_18_p

```
const numeric * GiNaC::_num_18_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.39 _ex_18

```
const ex GiNaC::_ex_18 = ex(*_num_18_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.40 _num_15_p

```
const numeric * GiNaC::_num_15_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.41 _ex_15

```
const ex GiNaC::_ex_15 = ex(*_num_15_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.42 _num_12_p

```
const numeric * GiNaC::_num_12_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.43 _ex_12

```
const ex GiNaC::_ex_12 = ex(*_num_12_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.44 _num_11_p

```
const numeric * GiNaC::_num_11_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.45 _ex_11

```
const ex GiNaC::_ex_11 = ex(*_num_11_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.46 _num_10_p

```
const numeric * GiNaC::_num_10_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.47 _ex_10

```
const ex GiNaC::_ex_10 = ex(*_num_10_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.48 _num_9_p

```
const numeric * GiNaC::_num_9_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.49 _ex_9

```
const ex GiNaC::_ex_9 = ex(*_num_9_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.50 _num_8_p

```
const numeric * GiNaC::_num_8_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.51 _ex_8

```
const ex GiNaC::_ex_8 = ex(*_num_8_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.52 _num_7_p

```
const numeric * GiNaC::_num_7_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.53 _ex_7

```
const ex GiNaC::_ex_7 = ex(*_num_7_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.54 _num_6_p

```
const numeric * GiNaC::_num_6_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.55 _ex_6

```
const ex GiNaC::_ex_6 = ex(*_num_6_p)
```

Referenced by GiNaC::su3d::eval_indexed(), GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.56 _num_5_p

```
const numeric * GiNaC::_num_5_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.57 _ex_5

```
const ex GiNaC::_ex_5 = ex(*_num_5_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.58 _num_4_p

```
const numeric * GiNaC::_num_4_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.59 _ex_4

```
const ex GiNaC::_ex_4 = ex(*_num_4_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.60 _num_3_p

```
const numeric * GiNaC::_num_3_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), and GiNaC::library_init::library_init().

7.1.4.61 _ex_3

```
const ex GiNaC::_ex_3 = ex(*_num_3_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.62 _num_2_p

```
const numeric * GiNaC::_num_2_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::library_init::library_init(), and tgamma_eval().

7.1.4.63 _ex_2

```
const ex GiNaC::_ex_2 = ex(*_num_2_p)
```

Referenced by GiNaC::diracgamma::contract_with(), GiNaC::spinmetric::contract_with(), GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.64 _num_1_p

```
const numeric * GiNaC::_num_1_p
```

Referenced by acos_conjugate(), asin_conjugate(), asinh_conjugate(), atan_conjugate(), atanh_conjugate(), beta_eval(), binomial(), GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::add::do_print_csrc(), doublefactorial(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::library_init::library_init(), operator-(), operator--(), operator--(), GiNaC::add::print_add(), GiNaC::mul::print_overall_coeff(), psi1_eval(), and psi2_eval().

7.1.4.65 _ex_1

```
const ex GiNaC::_ex_1 = ex(*_num_1_p)
```

Referenced by acos_eval(), acosh_deriv(), acosh_eval(), asin_eval(), atan2_deriv(), atan_deriv(), atan_eval(), atan_series(), atanh_deriv(), atanh_eval(), atanh_series(), GiNaC::diracgamma5::conjugate(), cos_eval(), GiNaC::power::derivative(), GiNaC::add::do_print_csrc(), GiNaC::mul::do_print_csrc(), GiNaC::power::do_print_csrc(), GiNaC::power::do_print_csrc_cl_N(), EllipticE_eval(), EllipticE_series(), EllipticK_series(), GiNaC::power::eval(), GiNaC::minkmetric::eval_indexed(), GiNaC::spinmetric::eval_indexed(), exminus(), exp_eval(), exp_power(), GiNaC::power::expand(), frac_cancel(), H_deriv(), H_eval(), lgamma_eval(), Li2_eval(), Li_eval(), GiNaC::library_init::library_init(), log_deriv(), log_expand(), log_series(), operator--(), operator--(), operator/(), operator/=(), psi1_series(), psi2_eval(), psi2_series(), replace_with_symbol(), sin_eval(), tan_eval(), tanh_eval(), GiNaC::power::to_polynomial(), GiNaC::ex::unit(), GiNaC::ex::unitcontprim(), and GiNaC::library_init::~library_init().

7.1.4.66 _num_1_2_p

```
const numeric * GiNaC::_num_1_2_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.67 _ex_1_2

```
const ex GiNaC::_ex_1_2 = ex(*_num_1_2_p)
```

Referenced by acos_deriv(), acos_eval(), acosh_deriv(), asin_deriv(), asin_eval(), asinh_deriv(), atan2_eval(), atan_series(), atanh_series(), cos_eval(), cosh_eval(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), Li2_eval(), GiNaC::library_init::library_init(), log_eval(), sin_eval(), sinh_eval(), tan_eval(), tanh_eval(), zeta1_eval(), and GiNaC::library_init::~library_init().

7.1.4.68 _num_1_3_p

```
const numeric * GiNaC::_num_1_3_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.69 _ex_1_3

```
const ex GiNaC::_ex_1_3 = ex(*_num_1_3_p)
```

Referenced by cos_eval(), GiNaC::su3d::eval_indexed(), GiNaC::library_init::library_init(), sin_eval(), and GiNaC::library_init::~library_init().

7.1.4.70 _num_1_4_p

```
const numeric * GiNaC::_num_1_4_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.71 _ex_1_4

```
const ex GiNaC::_ex_1_4 = ex(*_num_1_4_p)
```

Referenced by atan2_eval(), atan_eval(), cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and GiNaC::library_init::~library_init

7.1.4.72 _num0_p

```
const numeric * GiNaC::_num0_p
```

Referenced by GiNaC::numeric::add_dyn(), atan(), binomial(), GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), cos_eval(), divide_in_z(), GiNaC::power::eval(), exp_eval(), GiNaC::mul::expand(), fibonacci(), GiNaC::numeric::has(), GiNaC::add::integer_content(), iquo(), iquo(), irem(), irem(), isqrt(), Li2(), GiNaC::library_init::library_init(), mod(), GiNaC::relational::operator safe_bool(), GiNaC::numeric::power(), GiNaC::numeric::power_dyn(), sin_eval(), smod(), GiNaC::numeric::sub_dyn(), and tan_eval().

7.1.4.73 _ex0

```
const ex GiNaC::_ex0 = ex(*_num0_p)
```

Referenced by acos_eval(), acosh_eval(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::pseries::add_series(), asinh_eval(), atan2_eval(), atan_eval(), atan_series(), atanh_eval(), atanh_series(), base_and_index(), beta_eval(), binomial_sym(), GiNaC::relational::canonical(), GiNaC::add::coeff(), GiNaC::basic::coeff(), GiNaC::mul::coeff(), GiNaC::ncmul::coeff(), GiNaC::numeric::coeff(), GiNaC::power::coeff(), GiNaC::pseries::coeff(), GiNaC::basic::collect(), color_trace(), GiNaC::ex::content(), cos_eval(), csgn_series(), GiNaC::expairseq::default_overall_coeff(), GiNaC::basic::derivative(), GiNaC::constant::derivative(), GiNaC::idx::derivative(), GiNaC::indexed::derivative(), GiNaC::symbol::derivative(), GiNaC::matrix::determinant(), GiNaC::matrix::determinant_minor(), dirac_trace(), divide(), divide_in_z(), GiNaC::matrix::division_free_elimination(), EllipticE_eval(), EllipticE_series(), EllipticK_eval(), EllipticK_series(), eta_eval(), eta_evalf(), eta_series(), GiNaC::function::eval(), GiNaC::indexed::eval(), GiNaC::integral::eval(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::minkmetric::eval_indexed(), GiNaC::spinmetric::eval_indexed(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::tensdelta::eval_indexed(), GiNaC::tensepsilon::eval_indexed(), GiNaC::clifford::eval_ncmul(), GiNaC::indexed::expand(), find_common_factor(), GiNaC::matrix::fraction_free_elimination(), G2_eval(), G2_evalf(), G3_eval(), G3_evalf(), GiNaC::matrix::gauss_elimination(), gcd(), GINAC_IMPLEMENT_REGISTERED_CLAS GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), H_deriv(), H_eval(), GiNaC::ex::is_zero(), Li2_eval(), Li2_series(), Li_deriv(), Li_eval(), Li_evalf(), GiNaC::library_init::library_init(), log_eval(), log_series(), GiNaC::matrix::markowitz_elimination(), GiNaC::matrix::matrix(), GiNaC::matrix::matrix(), GiNaC::pseries::mul_series(), Order_eval(), GiNaC::pseries::power_const(), prem(), GiNaC::ex::primpart(), rem(), S_deriv(), S_eval(), GiNaC::basic::series(), GiNaC::ex::series(), GiNaC::symbol::series(), GiNaC::indexed::simplify_indexed_product, sin_eval(), sinh_eval(), sprem(), sqrfree(), sr_gcd(), step_series(), tan_eval(), tanh_eval(), GiNaC::ex::unitcontprim(), zeta1_deriv(), zeta1_eval(), zeta2_deriv(), and GiNaC::library_init::~library_init().

7.1.4.74 _num1_4_p

```
const numeric * GiNaC::_num1_4_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.75 _ex1_4

```
const ex GiNaC::_ex1_4 = ex(*_num1_4_p)
```

Referenced by atan2_eval(), atan_eval(), cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and GiNaC::library_init::~library_init

7.1.4.76 _num1_3_p

```
const numeric * GiNaC::_num1_3_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.77 _ex1_3

```
const ex GiNaC::_ex1_3 = ex(*_num1_3_p)
```

Referenced by acos_eval(), cos_eval(), GiNaC::su3d::eval_indexed(), GiNaC::library_init::library_init(), sin_eval(), tan_eval(), and GiNaC::library_init::~library_init().

7.1.4.78 _num1_2_p

```
const numeric * GiNaC::_num1_2_p
```

Referenced by GiNaC::library_init::library_init(), psi2_eval(), and tgamma_eval().

7.1.4.79 _ex1_2

```
const ex GiNaC::_ex1_2 = ex(*_num1_2_p)
```

Referenced by acos_eval(), asin_eval(), atan2_eval(), atan_series(), atanh_series(), cos_eval(), GiNaC::power::do_print_dflt(), GiNaC::power::do_print_latex(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::clifford::get_metric(), Li2_eval(), GiNaC::library_init::library_init(), log_eval(), psi1_eval(), psi2_eval(), sin_eval(), sqrt(), and GiNaC::library_init::~library_init().

7.1.4.80 _num1_p

```
const numeric * GiNaC::_num1_p
```

Referenced by acos_conjugate(), acosh_conjugate(), asin_conjugate(), asinh_conjugate(), atan(), atan_conjugate(), atanh_conjugate(), bernoulli(), binomial(), GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), GiNaC::numeric::denom(), GiNaC::numeric::div_dyn(), divide_in_z(), GiNaC::add::do_print_csrc(), doublefactorial(), GiNaC::mul::eval(), GiNaC::power::eval(), exp_eval(), GiNaC::power::expand_add(), frac_cancel(), gcd(), GiNaC::mul::info(), GiNaC::add::integer_content(), GiNaC::basic::integer_content(), lcm_of_coefficients_denominators(), lcmcoeff(), Li2_conjugate(), GiNaC::library_init::library_init(), GiNaC::basic::max_coefficient(), GiNaC::numeric::mul_dyn(), multiply_lcm(), operator++(), operator++(), GiNaC::matrix::pow(), GiNaC::numeric::power(), GiNaC::numeric::power_dyn(), GiNaC::add::print_add(), GiNaC::mul::print_overall_coeff(), psi2_eval(), GiNaC::add::recombine_pair_to_ex(), tgamma_eval(), and zeta1_eval().

7.1.4.81 _ex1

```
const ex GiNaC::_ex1 = ex(*_num1_p)
```

Referenced by acos_eval(), acosh_deriv(), acosh_eval(), GiNaC::pseries::add_series(), asin_eval(), asinh_deriv(), atan_deriv(), atan_eval(), atan_series(), atanh_deriv(), atanh_eval(), atanh_series(), base_and_index(), beta_eval(), binomial_sym(), GiNaC::basic::coeff(), GiNaC::ncmul::coeff(), GiNaC::power::coeff(), GiNaC::basic::collect(), color_trace(), GiNaC::add::combine_ex_with_coeff_to_pair(), GiNaC::mul::combine_ex_with_coeff_to_pair(), GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::mul::combine_pair_with_coeff_to_pair(), GiNaC::cliffordunit::contract_with(), GiNaC::diracgamma::contract_with(), GiNaC::matrix::contract_with(), GiNaC::spinmetric::contract_with(), GiNaC::su3d::contract_with(), GiNaC::su3f::contract_with(), GiNaC::su3t::contract_with(), GiNaC::tensepsilon::contract_with(), cos_eval(), cosh_eval(), GiNaC::mul::default_overall_coeff(), GiNaC::mul::derivative(), GiNaC::power::derivative(), GiNaC::symbol::derivative(), GiNaC::matrix::determinant_minor(), dirac_trace(), divide(), divide_in_z(), GiNaC::add::do_print_csrc(), GiNaC::mul::do_print_csrc(), EllipticE_eval(), EllipticE_series(), EllipticK_series(), GiNaC::mul::eval(), GiNaC::ncmul::eval(), GiNaC::power::eval(), GiNaC::minkmetric::eval_indexed(), GiNaC::spinmetric::eval_indexed(), GiNaC::tensdelta::eval_indexed(), GiNaC::mul::evalm(), exp_eval(), GiNaC::expairseq::expair_needs_further_processing(), GiNaC::mul::expair_needs_further_processi GiNaC::mul::expand(), GiNaC::ncmul::expand(), GiNaC::power::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add GiNaC::power::expand_mul(), find_common_factor(), frac_cancel(), GiNaC::matrix::fraction_free_elimination(), G2_eval(), G2_evalf(), G3_eval(), G3_evalf(), gcd(), gcd_pf_pow(), gcd_pf_pow_pow(), GINAC_IMPLEMENT_REGISTERED_CLASS GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), H_deriv(), H_eval(), GiNaC::ncmul::hold_ncmul, GiNaC::power::imag_part(), GiNaC::matrix::inverse(), lgamma_series(), Li2_deriv(), Li2_eval(), Li2_series(), Li_eval(), Li_evalf(), Li_series(),

GiNaC::library_init::library_init(), log_eval(), log_expand(), log_series(), GiNaC::expairseq::make_flat(), GiNaC::expairseq::make_flat(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::pseries::mul_series(), GiNaC::basic::normal(), GiNaC::power::normal(), GiNaC::pseries::normal(), GiNaC::symbol::normal(), operator++(), operator++(), Order_eval(), Order_series(), GiNaC::matrix::pow(), GiNaC::pseries::power_const(), prem(), GiNaC::ex::primpart(), GiNaC::pseries::print_series(), psi1_deriv(), psi1_series(), psi2_deriv(), psi2_eval(), psi2_series(), quo(), GiNaC::power::real_part(), GiNaC::mul::recombine_pair_to_ex(), GiNaC::tensor::replace_contr_index(), S_series(), GiNaC::add::series(), GiNaC::basic::series(), GiNaC::integral::series(), GiNaC::mul::series(), GiNaC::power::series(), GiNaC::pseries::series(), GiNaC::symbol::series(), GiNaC::indexed::simplify_indexed_product, sin_eval(), sinh_eval(), GiNaC::add::split_ex_to_pair(), GiNaC::expairseq::split_ex_to_pair(), GiNaC::mul::split_ex_to_pair(), sprem(), sqrfree_parfrac(), sqrfree_yun(), sr_gcd(), tan_deriv(), tan_eval(), tanh_deriv(), tanh_eval(), tgamma_series(), GiNaC::expairseq::to_polynomial(), GiNaC::expairseq::to_rational(), GiNaC::ex::unit(), unit_matrix(), GiNaC::ex::unitcontprim(), zeta1_deriv(), zeta2_deriv(), and GiNaC::library_init::∼library_init().

7.1.4.82 _num2_p

```
const numeric * GiNaC::_num2_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), exp_eval(), GiNaC::power::expand_add_2(), Li2_series(), GiNaC::library_init::library_init(), GiNaC::matrix::pow(), psi1_eval(), psi2_eval(), tgamma_eval(), and zeta1_eval().

7.1.4.83 _ex2

```
const ex GiNaC::_ex2 = ex(*_num2_p)
```

Referenced by acos_deriv(), asin_deriv(), asinh_deriv(), atan2_deriv(), atan_deriv(), atanh_deriv(), canonicalize_clifford(), GiNaC::diracgamma::contract_with(), GiNaC::spinmetric::contract_with(), cos_eval(), cosh_eval(), csgn_power(), epsilon_tensor(), exp_eval(), GiNaC::power::expand_add_2(), Li2_eval(), Li2_series(), Li_eval(), GiNaC::library_init::library_init(), product_to_exvector(), psi1_eval(), GiNaC::indexed::simplify_indexed, sin_eval(), sinh_eval(), tan_deriv(), tan_eval(), tanh_deriv(), tanh_eval(), and GiNaC::library_init::∼library_init().

7.1.4.84 _num3_p

```
const numeric * GiNaC::_num3_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), exp_eval(), and GiNaC::library_init::library_init().

7.1.4.85 _ex3

```
const ex GiNaC::_ex3 = ex(*_num3_p)
```

Referenced by color_trace(), cos_eval(), epsilon_tensor(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::library_init::library_init(), sin_eval(), tan_eval(), and GiNaC::library_init::∼library_init().

7.1.4.86 _num4_p

```
const numeric * GiNaC::_num4_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), exp_eval(), and GiNaC::library_init::library_init().

7.1.4.87 _ex4

```
const ex GiNaC::_ex4 = ex(*_num4_p)
```

Referenced by dirac_trace(), GiNaC::library_init::library_init(), lorentz_eps(), and GiNaC::library_init::~library_init().

7.1.4.88 _num5_p

```
const numeric * GiNaC::_num5_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and tan_eval().

7.1.4.89 _ex5

```
const ex GiNaC::_ex5 = ex(*_num5_p)
```

Referenced by cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and GiNaC::library_init::~library_init().

7.1.4.90 _num6_p

```
const numeric * GiNaC::_num6_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), GiNaC::library_init::library_init(), and sin_eval().

7.1.4.91 _ex6

```
const ex GiNaC::_ex6 = ex(*_num6_p)
```

Referenced by cos_eval(), Li2_eval(), GiNaC::library_init::library_init(), sin_eval(), and GiNaC::library_init::~library_init().

7.1.4.92 _num7_p

```
const numeric * GiNaC::_num7_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), and GiNaC::library_init::library_init().

7.1.4.93 _ex7

```
const ex GiNaC::_ex7 = ex(*_num7_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.94 _num8_p

```
const numeric * GiNaC::_num8_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), and GiNaC::library_init::library_init().

7.1.4.95 _ex8

```
const ex GiNaC::_ex8 = ex(*_num8_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.96 _num9_p

```
const numeric * GiNaC::_num9_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), and GiNaC::library_init::library_init().

7.1.4.97 _ex9

```
const ex GiNaC::_ex9 = ex(*_num9_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.98 _num10_p

```
const numeric * GiNaC::_num10_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and tan_eval().

7.1.4.99 _ex10

```
const ex GiNaC::_ex10 = ex(*_num10_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.100 _num11_p

```
const numeric * GiNaC::_num11_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), and GiNaC::library_init::library_init().

7.1.4.101 _ex11

```
const ex GiNaC::_ex11 = ex(*_num11_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.102 _num12_p

```
const numeric * GiNaC::_num12_p
```

Referenced by GiNaC::ex::construct_from_int(), GiNaC::ex::construct_from_long(), GiNaC::ex::construct_from_uint(), GiNaC::ex::construct_from_ulong(), cos_eval(), and GiNaC::library_init::library_init().

7.1.4.103 _ex12

```
const ex GiNaC::_ex12 = ex(*_num12_p)
```

Referenced by Li2_eval(), GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.104 _num15_p

```
const numeric * GiNaC::_num15_p
```

Referenced by cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and tan_eval().

7.1.4.105 _ex15

```
const ex GiNaC::_ex15 = ex(*_num15_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.106 _num18_p

```
const numeric * GiNaC::_num18_p
```

Referenced by GiNaC::library_init::library_init(), and sin_eval().

7.1.4.107 _ex18

```
const ex GiNaC::_ex18 = ex(*_num18_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.108 _num20_p

```
const numeric * GiNaC::_num20_p
```

Referenced by cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and tan_eval().

7.1.4.109 _ex20

```
const ex GiNaC::_ex20 = ex(*_num20_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.110 _num24_p

```
const numeric * GiNaC::_num24_p
```

Referenced by cos_eval(), and GiNaC::library_init::library_init().

7.1.4.111 _ex24

```
const ex GiNaC::_ex24 = ex(*_num24_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.112 _num25_p

```
const numeric * GiNaC::_num25_p
```

Referenced by cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and tan_eval().

7.1.4.113 _ex25

```
const ex GiNaC::_ex25 = ex(*_num25_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.114 _num30_p

```
const numeric * GiNaC::_num30_p
```

Referenced by cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and tan_eval().

7.1.4.115 _ex30

```
const ex GiNaC::_ex30 = ex(*_num30_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.116 _num48_p

```
const numeric * GiNaC::_num48_p
```

Referenced by GiNaC::library_init::library_init().

7.1.4.117 _ex48

```
const ex GiNaC::_ex48 = ex(*_num48_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.1.4.118 _num60_p

```
const numeric * GiNaC::_num60_p
```

Referenced by cos_eval(), GiNaC::library_init::library_init(), sin_eval(), and tan_eval().

7.1.4.119 _ex60

```
const ex GiNaC::_ex60 = ex(*_num60_p)
```

Referenced by cos_eval(), GiNaC::library_init::library_init(), sin_eval(), tan_eval(), and GiNaC::library_init::~library_init().

7.1.4.120 _num120_p

```
const numeric * GiNaC::_num120_p
```

Referenced by cos_eval(), GiNaC::library_init::library_init(), and sin_eval().

7.1.4.121 _ex120

```
const ex GiNaC::_ex120 = ex(*_num120_p)
```

Referenced by GiNaC::library_init::library_init(), and GiNaC::library_init::~library_init().

7.2 GiNaC::Gt_detail Namespace Reference

Classes

- class TransformExpressionWithCache
- class pathintegral_term
- class pathintegral_term_list

Functions

- std::vector< std::vector< int > > integer_partition (const int n, const int m)
- template<typename Kernel>
  ex deconcatenate_path (const std::vector< Kernel > &args, const std::vector< ex > &endpoints, const std::function< ex(std::vector< Kernel > &new_args, const ex &start, const ex &end)> &construct)
- template ex deconcatenate_path< ex > (const std::vector< ex > &args, const std::vector< ex > &endpoints, const std::function< ex(std::vector< ex > &new_args, const ex &start, const ex &end)> &construct)
- template ex deconcatenate_path< Gt::kernel > (const std::vector< Gt::kernel > &args, const std::vector< ex > &endpoints, const std::function< ex(std::vector< Gt::kernel > &new_args, const ex &start, const ex &end)> &construct)
- std::ostream & operator<< (std::ostream &os, const pathintegral_term &term)
- bool operator< (const pathintegral_term &a, const pathintegral_term &b)

7.2.1 Function Documentation

7.2.1.1 integer_partition()

```
std::vector< std::vector< int > > GiNaC::Gt_detail::integer_partition (
            const int n,
            const int m)
```

References integer_partition().

Referenced by deconcatenate_path(), and integer_partition().

7.2.1.2 deconcatenate_path()

```
template<typename Kernel>
ex GiNaC::Gt_detail::deconcatenate_path (
            const std::vector< Kernel > & args,
            const std::vector< ex > & endpoints,
            const std::function< ex(std::vector< Kernel > &new_args, const ex &start, const
ex &end)> & construct)
```

References GiNaC::add::add(), deconcatenate_path(), integer_partition(), and GiNaC::mul::mul().

Referenced by GiNaC::Gt::applyIntegrationPath(), deconcatenate_path(), deconcatenate_path< ex >(), and GiNaC::Gt_detail::pathintegral_term::G_path().

7.2.1.3 deconcatenate_path< ex >()

```
template ex GiNaC::Gt_detail::deconcatenate_path< ex > (
            const std::vector< ex > & args,
            const std::vector< ex > & endpoints,
            const std::function< ex(std::vector< ex > &new_args, const ex &start, const ex
&end)> & construct)
```

References deconcatenate_path().

7.2.1.4 deconcatenate_path< Gt::kernel >()

```
template ex GiNaC::Gt_detail::deconcatenate_path< Gt::kernel > (
             const std::vector< Gt::kernel > & args,
             const std::vector< ex > & endpoints,
             const std::function< ex(std::vector< Gt::kernel > &new_args, const ex &start,
const ex &end)> & construct)
```

7.2.1.5 operator<<()

```
std::ostream & GiNaC::Gt_detail::operator<< (
             std::ostream & os,
             const pathintegral_term & term)
```

7.2.1.6 operator<()

```
bool GiNaC::Gt_detail::operator< (
             const pathintegral_term & a,
             const pathintegral_term & b)
```

7.3 GiNaC::internal Namespace Reference

Classes

- struct _iter_rep

7.4 std Namespace Reference

Classes

- struct equal_to< GiNaC::ex >

  Specialization of std::equal_to() for ex objects.
- struct hash< GiNaC::ex >

  Specialization of std::hash() for ex objects.
- struct less< GiNaC::ptr< T > >

  Specialization of std::less for ptr<T> to enable ordering of ptr<T> objects (e.g.

Functions

- template<> void swap (GiNaC::ex &a, GiNaC::ex &b)

  Specialization of std::swap() for ex objects.

7.4.1 Function Documentation

7.4.1.1 swap()

```
template<>
void std::swap (
            GiNaC::ex & a,
            GiNaC::ex & b)  [inline]
```

Specialization of std::swap() for ex objects.

References GiNaC::ex::swap(), and swap().

Referenced by GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::matrix::markowitz_elimination(), GiNaC::permutation_sign(), and swap().

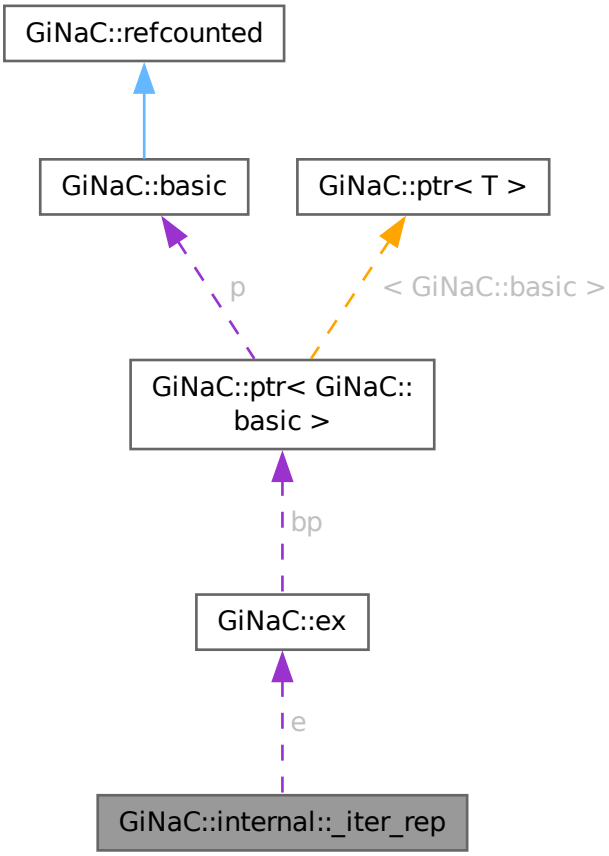
Chapter 8

Class Documentation

8.1 GiNaC::internal::_iter_rep Struct Reference

`#include <ex.h>`

Collaboration diagram for GiNaC::internal::_iter_rep:



Public Member Functions

- _iter_rep (const ex &e_, size_t i_, size_t i_end_)
- bool operator== (const _iter_rep &other) const noexcept
- bool operator!= (const _iter_rep &other) const noexcept

Public Attributes

- ex e
- size_t i
- size_t i_end

8.1.1 Constructor & Destructor Documentation

8.1.1.1 _iter_rep()

```
GiNaC::internal::_iter_rep::_iter_rep (
            const ex & e_,
            size_t i_,
            size_t i_end_)  [inline]
```

References e, i, and i_end.

Referenced by operator!=(), and operator==().

8.1.2 Member Function Documentation

8.1.2.1 operator==()

```
bool GiNaC::internal::_iter_rep::operator== (
            const _iter_rep & other) const  [inline], [noexcept]
```

References _iter_rep(), GiNaC::are_ex_trivially_equal(), e, and i.

8.1.2.2 operator"!=()

```
bool GiNaC::internal::_iter_rep::operator!= (
            const _iter_rep & other) const  [inline], [noexcept]
```

References _iter_rep().

8.1.3 Member Data Documentation

8.1.3.1 e

```
ex GiNaC::internal::_iter_rep::e
```

Referenced by _iter_rep(), GiNaC::const_postorder_iterator::descend(), GiNaC::const_preorder_iterator::increment(), and operator==().

8.1.3.2 i

`size_t GiNaC::internal::_iter_rep::i`

Referenced by _iter_rep(), GiNaC::const_postorder_iterator::descend(), GiNaC::const_preorder_iterator::increment(), and operator==().

8.1.3.3 i_end

`size_t GiNaC::internal::_iter_rep::i_end`

Referenced by _iter_rep(), and GiNaC::const_preorder_iterator::increment().

The documentation for this struct was generated from the following file:

- ex.h

8.2 GiNaC::_numeric_digits Class Reference

This class is used to instantiate a global singleton object Digits which behaves just like Maple's Digits.

`#include <numeric.h>`

Public Member Functions

- _numeric_digits ()
  _numeric_digits default ctor, checking for singleton invariance.
- _numeric_digits & operator= (long prec)
  Assign a native long to global Digits object.
- operator long ()
  Convert global Digits object to native type long.
- void print (std::ostream &os) const
  Append global Digits object to ostream.
- void add_callback (digits_changed_callback callback)
  Add a new callback function.

Private Attributes

- long digits
  Number of decimal digits.
- std::vector< digits_changed_callback > callbacklist

Static Private Attributes

- static bool too_late = false
  Already one object present.

8.2.1 Detailed Description

This class is used to instantiate a global singleton object Digits which behaves just like Maple's Digits.

We need an object rather than a dumber basic type since as a side-effect we let it change cl_default_float_format when it gets changed. The only other meaningful thing to do with it is converting it to an unsigned, for temporarily storing its value e.g. The user must not create an own working object of this class! Since C++ forces us to make the class definition visible in order to use an object we put in a flag which prevents other objects of that class to be created.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 _numeric_digits()

```
GiNaC::_numeric_digits::_numeric_digits ()
```

_numeric_digits default ctor, checking for singleton invariance.

References digits, and too_late.

Referenced by operator=().

8.2.3 Member Function Documentation

8.2.3.1 operator=()

```
_numeric_digits & GiNaC::_numeric_digits::operator= (
            long prec)
```

Assign a native long to global Digits object.

References _numeric_digits(), callbacklist, and digits.

8.2.3.2 operator long()

```
GiNaC::_numeric_digits::operator long ()
```

Convert global Digits object to native type long.

References digits.

8.2.3.3 print()

```
void GiNaC::_numeric_digits::print (
            std::ostream & os) const
```

Append global Digits object to ostream.

References digits.

Referenced by GiNaC::operator<<().

8.2.3.4 add_callback()

```
void GiNaC::_numeric_digits::add_callback (
             digits_changed_callback callback)
```

Add a new callback function.

References callbacklist.

8.2.4 Member Data Documentation

8.2.4.1 digits

```
long GiNaC::_numeric_digits::digits  [private]
```

Number of decimal digits.

Referenced by _numeric_digits(), operator long(), operator=(), and print().

8.2.4.2 too_late

```
bool GiNaC::_numeric_digits::too_late = false  [static], [private]
```

Already one object present.

Referenced by _numeric_digits().

8.2.4.3 callbacklist

```
std::vector<digits_changed_callback> GiNaC::_numeric_digits::callbacklist  [private]
```

Referenced by add_callback(), and operator=().

The documentation for this class was generated from the following files:

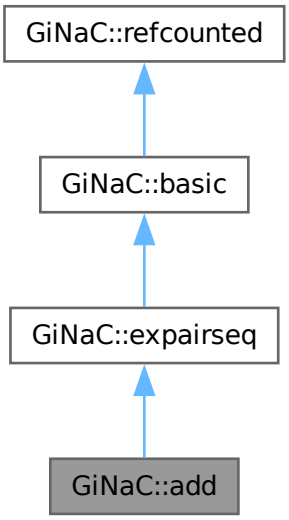
- numeric.h
- numeric.cpp

8.3 GiNaC::add Class Reference

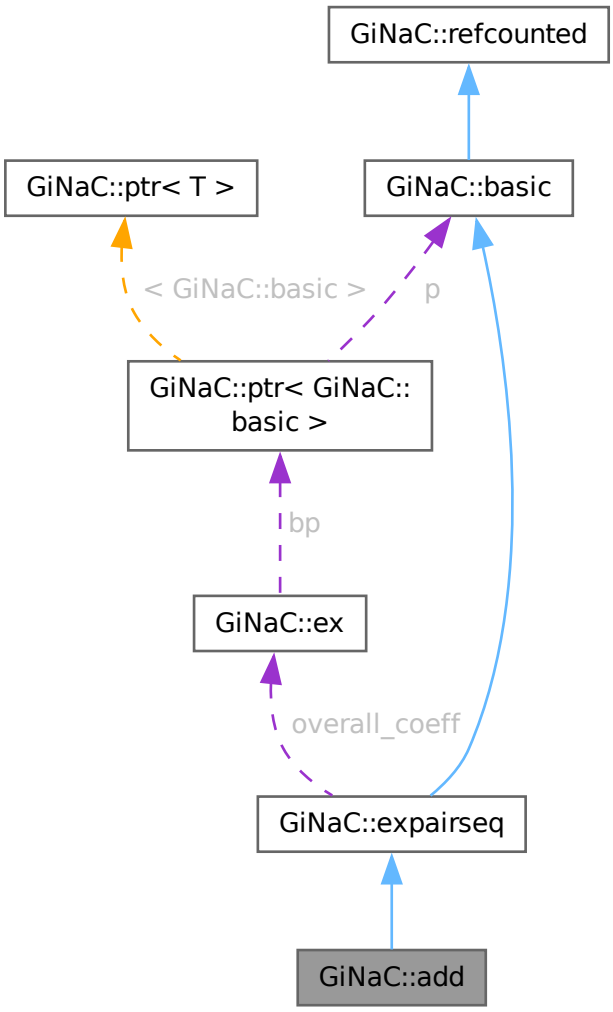
Sum of expressions.

`#include <add.h>`

Inheritance diagram for GiNaC::add:



Collaboration diagram for GiNaC::add:



Public Member Functions

- add (const ex &lh, const ex &rh)
- add (const exvector &v)
- add (const epvector &v)
- add (const epvector &v, const ex &oc)
- add (epvector &&v)
- add (epvector &&v, const ex &oc)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- bool is_polynomial (const ex &var) const override

  Check whether this is a polynomial in the given variables.

- int degree (const ex &s) const override

  Return degree of highest power in object s.
- int ldegree (const ex &s) const override

  Return degree of lowest power in object s.
- ex coeff (const ex &s, int n=1) const override

  Return coefficient of degree n in object s.
- ex eval () const override

  Perform automatic term rewriting rules in this class.
- ex evalm () const override

  Evaluate sums, products and integer powers of matrices.
- ex series (const relational &r, int order, unsigned options=0) const override

  Implementation of ex::series() for sums.
- ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const override

  Implementation of ex::normal() for a sum.
- numeric integer_content () const override
- ex smod (const numeric &xi) const override

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- numeric max_coefficient () const override

  Implementation ex::max_coefficient().
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- exvector get_free_indices () const override

  Return a vector containing the free indices of an expression.
- ex eval_ncmul (const exvector &v) const override

Public Member Functions inherited from GiNaC::expairseq

- expairseq (const ex &lh, const ex &rh)
- expairseq (const exvector &v)
- expairseq (const epvector &v, const ex &oc, bool do_index_renaming=false)
- expairseq (epvector &&vp, const ex &oc, bool do_index_renaming=false)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex map (map_function &f) const override

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- ex eval () const override

  Perform coefficient-wise automatic term rewriting rules in this class.
- ex to_rational (exmap &repl) const override

  Implementation of ex::to_rational() for expairseqs.
- ex to_polynomial (exmap &repl) const override

  Implementation of ex::to_polynomial() for expairseqs.
- bool match (const ex &pattern, exmap &repl_lst) const override

  Check whether the expression matches a given pattern.
- ex subs (const exmap &m, unsigned options=0) const override

Substitute a set of objects by arbitrary expressions.

- ex conjugate () const override
- void archive (archive_node &n) const override

  Save (serialize) the object into archive node.
- void read_archive (const archive_node &n, lst &syms) override

  Load (deserialize) the object from an archive node.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual void accept (GiNaC::visitor &v) const
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

Helper function for subs().

- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for a sum.
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- ex thisexpairseq (const epvector &v, const ex &oc, bool do_index_renaming=false) const override

  Create an object of this type.
- ex thisexpairseq (epvector &&vp, const ex &oc, bool do_index_renaming=false) const override
- expair split_ex_to_pair (const ex &e) const override

  Form an expair from an ex, using the corresponding semantics.
- expair combine_ex_with_coeff_to_pair (const ex &e, const ex &c) const override
- expair combine_pair_with_coeff_to_pair (const expair &p, const ex &c) const override
- ex recombine_pair_to_ex (const expair &p) const override

  Form an ex out of an expair, using the corresponding semantics.
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- void print_add (const print_context &c, const char *openbrace, const char *closebrace, const char *mul_sym, unsigned level) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_csrc (const print_csrc &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::expairseq

- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- unsigned return_type () const override
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- virtual void printseq (const print_context &c, char delim, unsigned this_precedence, unsigned upper_precedence) const
- virtual void printpair (const print_context &c, const expair &p, unsigned upper_precedence) const
- virtual bool expair_needs_further_processing (epp it)
- virtual ex default_overall_coeff () const
- virtual void combine_overall_coeff (const ex &c)
- virtual void combine_overall_coeff (const ex &c1, const ex &c2)
- virtual bool can_make_flat (const expair &p) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void construct_from_2_ex (const ex &lh, const ex &rh)
- void construct_from_2_expairseq (const expairseq &s1, const expairseq &s2)
- void construct_from_expairseq_ex (const expairseq &s, const ex &e)
- void construct_from_exvector (const exvector &v)
- void construct_from_epvector (const epvector &v, bool do_index_renaming=false)
- void construct_from_epvector (epvector &&v, bool do_index_renaming=false)
- void make_flat (const exvector &v)

  Combine this expairseq with argument exvector.
- void make_flat (const epvector &v, bool do_index_renaming=false)

  Combine this expairseq with argument epvector.
- void canonicalize ()

  Brings this expairseq into a sorted (canonical) form.
- void combine_same_terms_sorted_seq ()

  Compact a presorted expairseq by combining all matching expairs to one each.
- bool is_canonical () const

  Check if this expairseq is in sorted (canonical) form.
- epvector expandchildren (unsigned options) const

  Member-wise expand the expairs in this sequence.
- epvector evalchildren () const

  Member-wise evaluate the expairs in this sequence.
- epvector subschildren (const exmap &m, unsigned options=0) const

  Member-wise substitute in this sequence.

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

Default output to stream.

- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.

- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Friends

- class mul
- class power

Additional Inherited Members

Protected Attributes inherited from GiNaC::expairseq

- epvector seq
- ex overall_coeff

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags

- unsigned hashvalue

  hash value

8.3.1 Detailed Description

Sum of expressions.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 add() [1/6]

```
GiNaC::add::add (
            const ex & lh,
            const ex & rh)
```

References GiNaC::_ex0, GiNaC::expairseq::construct_from_2_ex(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

Referenced by GiNaC::Gt_detail::pathintegral_term_list::add(), GiNaC::Gt_detail::pathintegral_term_list::add(), GiNaC::scalar_products::add_vectors(), conjugate(), GiNaC::expairseq::construct_from_2_expairseq(), GiNaC::expairseq::construct_
GiNaC::Gt_detail::deconcatenate_path(), GiNaC::integral::expand(), GiNaC::Gt_detail::TransformExpressionWithCache< Type >::im
GiNaC::log_expand(), power, GiNaC::Gt::qExpand(), GiNaC::symmetry::read_archive(), GiNaC::Gt::regularise(), GiNaC::symmetry::symmetry(), GiNaC::Gt::tauToFundamental(), GiNaC::symmetry::validate(), and GiNaC::Gt::zisToFundamental().

8.3.2.2 add() [2/6]

```
GiNaC::add::add (
            const exvector & v)
```

References GiNaC::_ex0, GiNaC::expairseq::construct_from_exvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.3.2.3 add() [3/6]

```
GiNaC::add::add (
            const epvector & v)
```

References GiNaC::_ex0, GiNaC::expairseq::construct_from_epvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.3.2.4 add() [4/6]

```
GiNaC::add::add (
            const epvector & v,
            const ex & oc)
```

References GiNaC::expairseq::construct_from_epvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.3.2.5 add() [5/6]

```
GiNaC::add::add (
            epvector && v)
```

References GiNaC::_ex0, GiNaC::expairseq::construct_from_epvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.3.2.6 add() [6/6]

```
GiNaC::add::add (
            epvector && v,
            const ex & oc)
```

References GiNaC::expairseq::construct_from_epvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.3.3 Member Function Documentation

8.3.3.1 precedence()

```
unsigned GiNaC::add::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

References precedence().

Referenced by do_print_csrc(), precedence(), and print_add().

8.3.3.2 info()

```
bool GiNaC::add::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References GiNaC::info_flags::cinteger, GiNaC::info_flags::cinteger_polynomial, GiNaC::info_flags::crational, GiNaC::info_flags::crational_polynomial, GiNaC::info_flags::even, info(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::info_flags::integer_polynomial, GiNaC::ex::is_zero(), GiNaC::info_flags::nonnegative, GiNaC::info_flags::nonnegint, GiNaC::expairseq::overall_coeff, GiNaC::info_flags::polynomial, GiNaC::info_flags::posint, GiNaC::info_flags::positive, GiNaC::info_flags::rational, GiNaC::info_flags::rational_function, GiNaC::info_flags::rational_polynomial, GiNaC::info_flags::real, recombine_pair_to_ex(), and GiNaC::expairseq::seq.

Referenced by info().

8.3.3.3 is_polynomial()

```
bool GiNaC::add::is_polynomial (
            const ex & var) const  [override], [virtual]
```

Check whether this is a polynomial in the given variables.

Reimplemented from GiNaC::basic.

References GiNaC::expairseq::seq.

8.3.3.4 degree()

```
int GiNaC::add::degree (
            const ex & s) const  [override], [virtual]
```

Return degree of highest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, and GiNaC::expairseq::seq.

8.3.3.5 ldegree()

```
int GiNaC::add::ldegree (
            const ex & s) const  [override], [virtual]
```

Return degree of lowest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, and GiNaC::expairseq::seq.

8.3.3.6 coeff()

```
ex GiNaC::add::coeff (
             const ex & s,
             int n = 1) const  [override], [virtual]
```

Return coefficient of degree n in object s.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::clifford_max_label(), GiNaC::ex::coeff(), GiNaC::dirac_ONE(), GiNaC::dynallocate(), GiNaC::expair::expair(), GiNaC::ex::is_zero(), GiNaC::ncmul::ncmul(), GiNaC::expairseq::overall_coeff, and GiNaC::expairseq::seq.

Referenced by print_add(), and smod().

8.3.3.7 eval()

```
ex GiNaC::add::eval () const  [override], [virtual]
```

Perform automatic term rewriting rules in this class.

In the following x stands for a symbolic variables of type ex and c stands for such an expression that contain a plain number.

- +(;c) -> c
- +(x;0) -> x

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative, GiNaC::dynallocate(), GiNaC::expairseq::evalchildren(), GiNaC::status_flags::evaluat GiNaC::basic::flags, GINAC_ASSERT, GiNaC::basic::hold(), GiNaC::is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), GiNaC::expairseq::seq, and unlikely.

8.3.3.8 evalm()

```
ex GiNaC::add::evalm () const  [override], [virtual]
```

Evaluate sums, products and integer powers of matrices.

Reimplemented from GiNaC::basic.

References GiNaC::matrix::add(), GiNaC::dynallocate(), GiNaC::ex::evalm(), GiNaC::ex_to(), GiNaC::is_a(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), GiNaC::expairseq::seq, and split_ex_to_pair().

8.3.3.9 series()

```
ex GiNaC::add::series (
            const relational & r,
            int order,
            unsigned options = 0) const  [override], [virtual]
```

Implementation of ex::series() for sums.

This performs series addition when adding pseries objects.

See also

ex::series

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::ex_to(), GiNaC::is_exactly_a(), GiNaC::expairseq::op(), GiNaC::expairseq::overall_coeff, GiNaC::expairseq::seq, and GiNaC::ex::series().

8.3.3.10 normal()

```
ex GiNaC::add::normal (
            exmap & repl,
            exmap & rev_lookup,
            lst & modifier) const  [override], [virtual]
```

Implementation of ex::normal() for a sum.

It expands terms and performs fractional addition.

See also

ex::normal

Reimplemented from GiNaC::basic.

References GiNaC::ex_to(), expand(), GiNaC::frac_cancel(), GiNaC::gcd(), GINAC_ASSERT, GiNaC::subs_options::no_pattern, GiNaC::container< C >::nops(), GiNaC::ex::normal(), GiNaC::container< C >::op(), GiNaC::ex::op(), GiNaC::expairseq::overall_coef recombine_pair_to_ex(), and GiNaC::expairseq::seq.

8.3.3.11 integer_content()

```
numeric GiNaC::add::integer_content () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::_num0_p, GiNaC::_num1_p, GiNaC::denom(), GiNaC::ex_to(), GiNaC::gcd(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::lcm(), GiNaC::numer(), GiNaC::expairseq::overall_coeff, and GiNaC::expairseq::seq.

8.3.3.12 smod()

```
ex GiNaC::add::smod (
            const numeric & xi) const  [override], [virtual]
```

Apply symmetric modular homomorphism to an expanded multivariate polynomial.

This function is usually used internally by heur_gcd().

Parameters

xi	modulus

Returns

mapped polynomial

See also

heur_gcd

Reimplemented from GiNaC::basic.

References coeff(), GiNaC::dynallocate(), GiNaC::ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, GiNaC::expairseq::seq, and GiNaC::smod().

8.3.3.13 max_coefficient()

```
numeric GiNaC::add::max_coefficient () const  [override], [virtual]
```

Implementation ex::max_coefficient().

See also

heur_gcd

Reimplemented from GiNaC::basic.

References GiNaC::abs(), GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::expairseq::overall_coeff, and GiNaC::expairseq::seq.

8.3.3.14 conjugate()

```
ex GiNaC::add::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References add(), GiNaC::are_ex_trivially_equal(), conjugate(), GiNaC::ex::conjugate(), GiNaC::expairseq::nops(), and GiNaC::expairseq::op().

Referenced by conjugate().

8.3.3.15 real_part()

```
ex GiNaC::add::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::ex::coeff(), GiNaC::dynallocate(), GiNaC::expair::expair(), GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, GiNaC::info_flags::real, GiNaC::ex::real_part(), recombine_pair_to_ex(), GiNaC::expairseq::seq, and split_ex_to_pair().

8.3.3.16 imag_part()

```
ex GiNaC::add::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::ex::coeff(), GiNaC::dynallocate(), GiNaC::expair::expair(), GiNaC::ex::imag_part(), GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, GiNaC::info_flags::real, recombine_pair_to_ex(), GiNaC::expairseq::seq, and split_ex_to_pair().

8.3.3.17 get_free_indices()

```
exvector GiNaC::add::get_free_indices () const  [override], [virtual]
```

Return a vector containing the free indices of an expression.

Reimplemented from GiNaC::basic.

References GiNaC::ex::get_free_indices(), GiNaC::indices_consistent(), GiNaC::expairseq::nops(), and GiNaC::expairseq::op().

8.3.3.18 eval_ncmul()

```
ex GiNaC::add::eval_ncmul (
            const exvector & v) const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::expairseq::seq.

8.3.3.19 derivative()

```
ex GiNaC::add::derivative (
            const symbol & y) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for a sum.

It differentiates each term.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate(), GiNaC::expair::expair(), and GiNaC::expairseq::seq.

8.3.3.20 return_type()

```
unsigned GiNaC::add::return_type () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative, and GiNaC::expairseq::seq.

8.3.3.21 return_type_tinfo()

```
return_type_t GiNaC::add::return_type_tinfo () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::make_return_type_t(), and GiNaC::expairseq::seq.

8.3.3.22 thisexpairseq() [1/2]

```
ex GiNaC::add::thisexpairseq (
             const epvector & v,
             const ex & oc,
             bool do_index_renaming = false) const  [override], [protected], [virtual]
```

Create an object of this type.

This method works similar to a constructor. It is useful because expairseq has (at least) two possible different semantics but we want to inherit methods thus avoiding code duplication. Sometimes a method in expairseq has to create a new one of the same semantics, which cannot be done by a ctor because the name (add, mul,...) is unknown on the expairseq level. In order for this trick to work a derived class must of course override this definition.

Reimplemented from GiNaC::expairseq.

References GiNaC::dynallocate().

8.3.3.23 thisexpairseq() [2/2]

```
ex GiNaC::add::thisexpairseq (
             epvector && vp,
             const ex & oc,
             bool do_index_renaming = false) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::dynallocate().

8.3.3.24 split_ex_to_pair()

```
expair GiNaC::add::split_ex_to_pair (
            const ex & e) const  [override], [protected], [virtual]
```

Form an expair from an ex, using the corresponding semantics.

See also

expairseq::recombine_pair_to_ex()

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1, GiNaC::basic::clearflag(), GiNaC::dynallocate(), GiNaC::status_flags::evaluated, GiNaC::ex_to(), GiNaC::expair::expair(), GiNaC::status_flags::hash_calculated, GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), mul, and GiNaC::expairseq::overall_coeff.

Referenced by evalm(), imag_part(), and real_part().

8.3.3.25 combine_ex_with_coeff_to_pair()

```
expair GiNaC::add::combine_ex_with_coeff_to_pair (
            const ex & e,
            const ex & c) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1, GiNaC::basic::clearflag(), GiNaC::dynallocate(), GiNaC::status_flags::evaluated, GiNaC::ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, GiNaC::status_flags::hash_calculated, GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), likely, mul, and GiNaC::expairseq::overall_coeff.

8.3.3.26 combine_pair_with_coeff_to_pair()

```
expair GiNaC::add::combine_pair_with_coeff_to_pair (
            const expair & p,
            const ex & c) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1, GiNaC::_num1_p, GiNaC::expair::coeff, GiNaC::ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, GiNaC::is_exactly_a(), and GiNaC::expair::rest.

Referenced by GiNaC::mul::eval(), and GiNaC::power::expand_add_2().

8.3.3.27 recombine_pair_to_ex()

```
ex GiNaC::add::recombine_pair_to_ex (
            const expair & p) const  [override], [protected], [virtual]
```

Form an ex out of an expair, using the corresponding semantics.

See also

expairseq::split_ex_to_pair()

Reimplemented from GiNaC::expairseq.

References GiNaC::_num1_p, GiNaC::expair::coeff, GiNaC::dynallocate(), GiNaC::ex_to(), and GiNaC::expair::rest.

Referenced by eval(), evalm(), GiNaC::power::expand(), imag_part(), info(), normal(), and real_part().

8.3.3.28 expand()

```
ex GiNaC::add::expand (
            unsigned options = 0) const  [override], [protected], [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate(), GiNaC::expairseq::expandchildren(), GiNaC::status_flags::expanded, GiNaC::expairseq::overall_co and GiNaC::basic::setflag().

Referenced by normal().

8.3.3.29 print_add()

```
void GiNaC::add::print_add (
            const print_context & c,
            const char * openbrace,
            const char * closebrace,
            const char * mul_sym,
            unsigned level) const  [protected]
```

References GiNaC::_num1_p, GiNaC::_num_1_p, GiNaC::class_info< OPT >::first, coeff(), GiNaC::ex_to(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, precedence(), GiNaC::basic::print(), GiNaC::ex::print(), GiNaC::print_context::s, and GiNaC::expairseq::seq.

Referenced by do_print(), and do_print_latex().

8.3.3.30 do_print()

```
void GiNaC::add::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References print_add().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.3.3.31 do_print_latex()

```
void GiNaC::add::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

References print_add().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.3.3.32 do_print_csrc()

```
void GiNaC::add::do_print_csrc (
             const print_csrc & c,
             unsigned level) const  [protected]
```

References GiNaC::_ex1, GiNaC::_ex_1, GiNaC::_num1_p, GiNaC::_num_1_p, GiNaC::ex_to(), GiNaC::ex::info(), GiNaC::is_a(), GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, GiNaC::info_flags::positive, precedence(), GiNaC::ex::print(), GiNaC::info_flags::real, GiNaC::print_context::s, and GiNaC::expairseq::seq.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.3.3.33 do_print_python_repr()

```
void GiNaC::add::do_print_python_repr (
             const print_python_repr & c,
             unsigned level) const  [protected]
```

References GiNaC::expairseq::nops(), GiNaC::expairseq::op(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.3.4 Friends And Related Symbol Documentation

8.3.4.1 mul

```
friend class mul  [friend]
```

References mul.

Referenced by combine_ex_with_coeff_to_pair(), mul, and split_ex_to_pair().

8.3.4.2 power

```
friend class power  [friend]
```

References add(), and power.

Referenced by power.

The documentation for this class was generated from the following files:

- add.h
- add.cpp
- indexed.cpp
- normal.cpp
- pseries.cpp

8.4 GiNaC::archive Class Reference

This class holds archived versions of GiNaC expressions (class ex).

`#include <archive.h>`

Classes

- struct archived_ex

  Archived expression descriptor.

Public Member Functions

- archive ()
- ~archive ()
- archive (const ex &e)

  Construct archive from expression using the default name "ex".
- archive (const ex &e, const char *n)

  Construct archive from expression using the specified name.
- void archive_ex (const ex &e, const char *name)

  Archive an expression.
- ex unarchive_ex (const lst &sym_lst, const char *name) const

  Retrieve expression from archive by name.
- ex unarchive_ex (const lst &sym_lst, unsigned index=0) const

  Retrieve expression from archive by index.
- ex unarchive_ex (const lst &sym_lst, std::string &name, unsigned index=0) const

  Retrieve expression and its name from archive by index.
- unsigned num_expressions () const

  Return number of archived expressions.
- const archive_node & get_top_node (unsigned index=0) const

  Return reference to top node of an expression specified by index.
- void clear ()

  Clear all archived expressions.
- archive_node_id add_node (const archive_node &n)

  Add archive_node to archive if the corresponding expression is not already archived.
- archive_node & get_node (archive_node_id id)

  Retrieve archive_node by ID.
- void forget ()

  Delete cached unarchived expressions in all archive_nodes (mainly for debugging).
- void printraw (std::ostream &os) const

  Print archive to stream in ugly raw format (for debugging).
- archive_atom atomize (const std::string &s) const

  Atomize a string (i.e.
- const std::string & unatomize (archive_atom id) const

  Unatomize a string (i.e.

Private Types

- typedef std::map< std::string, archive_atom >::const_iterator inv_at_cit

  The map of from strings to indices of the atoms vectors allows for faster archiving.

Private Attributes

- std::vector< archive_node > nodes

  Vector of archived nodes.
- std::vector< archived_ex > exprs

  Vector of archived expression descriptors.
- std::vector< std::string > atoms

  Vector of atomized strings (using a vector allows faster unarchiving).
- std::map< std::string, archive_atom > inverse_atoms
- std::map< ex, archive_node_id, ex_is_less > exprtable

  Map of stored expressions to nodes for faster archiving.

Friends

- std::ostream & operator<< (std::ostream &os, const archive &ar)

  Write archive to binary data stream.
- std::istream & operator>> (std::istream &is, archive &ar)

  Read archive from binary data stream.

8.4.1 Detailed Description

This class holds archived versions of GiNaC expressions (class ex).

An archive can be constructed from an expression and then written to a stream; or it can be read from a stream and then unarchived, yielding back the expression. Archives can hold multiple expressions which can be referred to by name or index number. The main component of the archive class is a vector of archive_nodes which each store one object of class basic (or a derived class).

8.4.2 Member Typedef Documentation

8.4.2.1 inv_at_cit

```
typedef std::map<std::string,archive_atom>::const_iterator GiNaC::archive::inv_at_cit  [private]
```

The map of from strings to indices of the atoms vectors allows for faster archiving.

8.4.3 Constructor & Destructor Documentation

8.4.3.1 archive() [1/3]

```
GiNaC::archive::archive ()  [inline]
```

Referenced by operator<<, and operator>>.

8.4.3.2 ~archive()

```
GiNaC::archive::~archive ()  [inline]
```

8.4.3.3 archive() [2/3]

```
GiNaC::archive::archive (
            const ex & e)  [inline]
```

Construct archive from expression using the default name "ex".

References archive_ex().

8.4.3.4 archive() [3/3]

```
GiNaC::archive::archive (
            const ex & e,
            const char * n)  [inline]
```

Construct archive from expression using the specified name.

References archive_ex().

8.4.4 Member Function Documentation

8.4.4.1 archive_ex()

```
void GiNaC::archive::archive_ex (
            const ex & e,
            const char * name)
```

Archive an expression.

Parameters

e	the expression to be archived
name	name under which the expression is stored

References add_node(), GiNaC::archive_node::archive_node(), GiNaC::archive::archived_ex::archived_ex(), atomize(), and exprs.

Referenced by archive(), and archive().

8.4.4.2 unarchive_ex() [1/3]

```
ex GiNaC::archive::unarchive_ex (
            const lst & sym_lst,
            const char * name) const
```

Retrieve expression from archive by name.

Parameters

sym_lst	list of pre-defined symbols
name	name of expression

References atomize(), exprs, and nodes.

8.4.4.3 unarchive_ex() [2/3]

```
ex GiNaC::archive::unarchive_ex (
            const lst & sym_lst,
            unsigned index = 0) const
```

Retrieve expression from archive by index.

Parameters

sym_lst	list of pre-defined symbols
index	index of expression

See also

count_expressions

References exprs, and nodes.

8.4.4.4 unarchive_ex() [3/3]

```
ex GiNaC::archive::unarchive_ex (
            const lst & sym_lst,
            std::string & name,
            unsigned index = 0) const
```

Retrieve expression and its name from archive by index.

Parameters

sym_lst	list of pre-defined symbols
name	receives the name of the expression
index	index of expression

See also

count_expressions

References exprs, nodes, and unatomize().

8.4.4.5 num_expressions()

```
unsigned GiNaC::archive::num_expressions () const
```

Return number of archived expressions.

References exprs.

8.4.4.6 get_top_node()

```
const archive_node & GiNaC::archive::get_top_node (
            unsigned index = 0) const
```

Return reference to top node of an expression specified by index.

References exprs, and nodes.

8.4.4.7 clear()

```
void GiNaC::archive::clear ()
```

Clear all archived expressions.

References atoms, exprs, exprtable, inverse_atoms, and nodes.

8.4.4.8 add_node()

```
archive_node_id GiNaC::archive::add_node (
            const archive_node & n)
```

Add archive_node to archive if the corresponding expression is not already archived.

Returns

ID of archived node

References exprtable, GiNaC::archive_node::get_ex(), GiNaC::archive_node::has_ex(), and nodes.

Referenced by GiNaC::archive_node::add_ex(), and archive_ex().

8.4.4.9 get_node()

```
archive_node & GiNaC::archive::get_node (
            archive_node_id id)
```

Retrieve archive_node by ID.

References nodes.

Referenced by GiNaC::archive_node::find_ex(), GiNaC::archive_node::find_ex_by_loc(), and GiNaC::archive_node::find_ex_node().

8.4.4.10 forget()

```
void GiNaC::archive::forget ()
```

Delete cached unarchived expressions in all archive_nodes (mainly for debugging).

References GiNaC::archive_node::forget(), and nodes.

8.4.4.11 printraw()

```
void GiNaC::archive::printraw (
            std::ostream & os) const
```

Print archive to stream in ugly raw format (for debugging).

References atoms, exprs, nodes, and unatomize().

8.4.4.12 atomize()

```
archive_atom GiNaC::archive::atomize (
            const std::string & s) const
```

Atomize a string (i.e.

convert it into an ID number that uniquely represents the string).

References atoms, and inverse_atoms.

Referenced by GiNaC::archive_node::add_bool(), GiNaC::archive_node::add_ex(), GiNaC::archive_node::add_string(), GiNaC::archive_node::add_unsigned(), archive_ex(), GiNaC::archive_node::find_bool(), GiNaC::archive_node::find_ex(), GiNaC::archive_node::find_ex_node(), GiNaC::archive_node::find_first(), GiNaC::archive_node::find_last(), GiNaC::archive_node::find_property_range(), GiNaC::archive_node::find_string(), GiNaC::archive_node::find_unsigned(), and unarchive_ex().

8.4.4.13 unatomize()

```
const std::string & GiNaC::archive::unatomize (
            archive_atom id) const
```

Unatomize a string (i.e.

convert the ID number back to the string).

References atoms.

Referenced by GiNaC::archive_node::find_string(), GiNaC::archive_node::get_properties(), printraw(), GiNaC::archive_node::printraw and unarchive_ex().

8.4.5 Friends And Related Symbol Documentation

8.4.5.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const archive & ar)  [friend]
```

Write archive to binary data stream.

References archive(), atoms, exprs, GINACLIB_ARCHIVE_VERSION, nodes, operator<<, and GiNaC::write_unsigned().

Referenced by operator<<.

8.4.5.2 operator>>

```
std::istream & operator>> (
            std::istream & is,
            archive & ar)  [friend]
```

Read archive from binary data stream.

References archive(), atoms, exprs, GINACLIB_ARCHIVE_AGE, GINACLIB_ARCHIVE_VERSION, inverse_atoms, nodes, operator>>, and GiNaC::read_unsigned().

Referenced by operator>>.

8.4.6 Member Data Documentation

8.4.6.1 nodes

```
std::vector<archive_node> GiNaC::archive::nodes  [private]
```

Vector of archived nodes.

Referenced by add_node(), clear(), forget(), get_node(), get_top_node(), operator<<, operator>>, printraw(), unarchive_ex(), unarchive_ex(), and unarchive_ex().

8.4.6.2 exprs

```
std::vector<archived_ex> GiNaC::archive::exprs  [private]
```

Vector of archived expression descriptors.

Referenced by archive_ex(), clear(), get_top_node(), num_expressions(), operator<<, operator>>, printraw(), unarchive_ex(), unarchive_ex(), and unarchive_ex().

8.4.6.3 atoms

```
std::vector<std::string> GiNaC::archive::atoms  [mutable], [private]
```

Vector of atomized strings (using a vector allows faster unarchiving).

Referenced by atomize(), clear(), operator<<, operator>>, printraw(), and unatomize().

8.4.6.4 inverse_atoms

```
std::map<std::string, archive_atom> GiNaC::archive::inverse_atoms  [mutable], [private]
```

Referenced by atomize(), clear(), and operator>>.

8.4.6.5 exprtable

```
std::map<ex, archive_node_id, ex_is_less> GiNaC::archive::exprtable  [mutable], [private]
```

Map of stored expressions to nodes for faster archiving.

Referenced by add_node(), and clear().

The documentation for this class was generated from the following files:

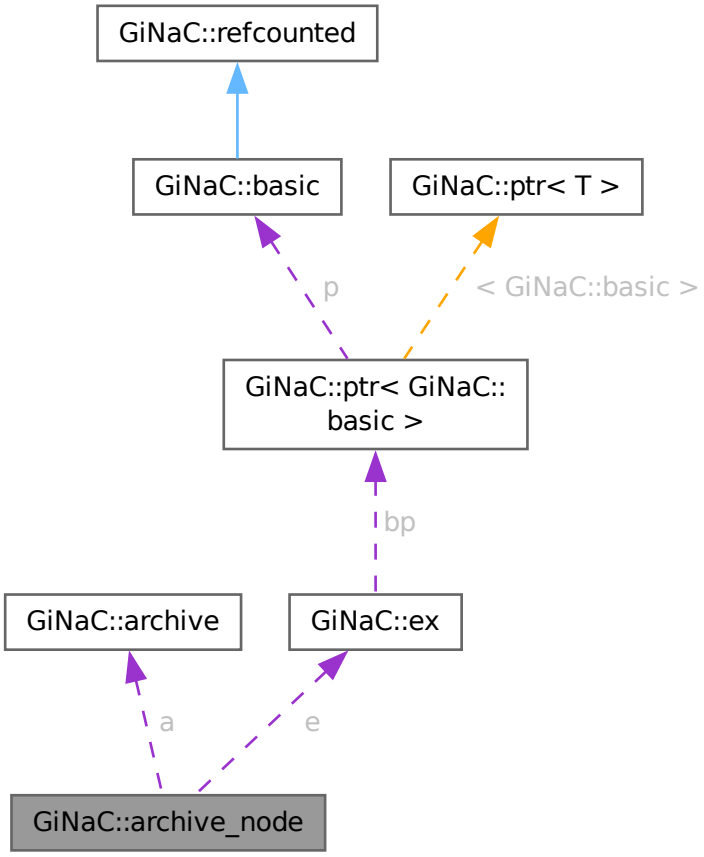
- archive.h
- archive.cpp

8.5 GiNaC::archive_node Class Reference

This class stores all properties needed to record/retrieve the state of one object of class basic (or a derived class).

```
#include <archive.h>
```

Collaboration diagram for GiNaC::archive_node:



Classes

- struct property_info

  Information about a stored property.
- struct property

  Archived property (data type, name and associated data).
- struct archive_node_cit_range

Public Types

- enum property_type { PTYPE_BOOL , PTYPE_UNSIGNED , PTYPE_STRING , PTYPE_NODE }

  Property data types.
- typedef std::vector< property_info > propinfovector
- typedef std::vector< property >::const_iterator archive_node_cit

Public Member Functions

- archive_node (archive &ar)
- archive_node (archive &ar, const ex &expr)

  Recursively construct archive node from expression.
- const archive_node & operator= (const archive_node &other)

  Assignment operator of archive_node.
- void add_bool (const std::string &name, bool value)

  Add property of type "bool" to node.
- void add_unsigned (const std::string &name, unsigned value)

  Add property of type "unsigned int" to node.
- void add_string (const std::string &name, const std::string &value)

  Add property of type "string" to node.
- void add_ex (const std::string &name, const ex &value)

  Add property of type "ex" to node.
- bool find_bool (const std::string &name, bool &ret, unsigned index=0) const

  Retrieve property of type "bool" from node.
- bool find_unsigned (const std::string &name, unsigned &ret, unsigned index=0) const

  Retrieve property of type "unsigned" from node.
- bool find_string (const std::string &name, std::string &ret, unsigned index=0) const

  Retrieve property of type "string" from node.
- archive_node_cit find_first (const std::string &name) const

  Find the location in the vector of properties of the first/last property with a given name.
- archive_node_cit find_last (const std::string &name) const
- archive_node_cit_range find_property_range (const std::string &name1, const std::string &name2) const

  Find a range of locations in the vector of properties.
- bool find_ex (const std::string &name, ex &ret, lst &sym_lst, unsigned index=0) const

  Retrieve property of type "ex" from node.
- void find_ex_by_loc (archive_node_cit loc, ex &ret, lst &sym_lst) const

  Retrieve property of type "ex" from the node if it is known that this node in fact contains such a property at the given location.
- const archive_node & find_ex_node (const std::string &name, unsigned index=0) const

  Retrieve property of type "ex" from node, returning the node of the sub-expression.
- void get_properties (propinfovector &v) const

  Return vector of properties stored in node.
- ex unarchive (lst &sym_lst) const

  Convert archive node to GiNaC expression.
- bool has_same_ex_as (const archive_node &other) const

  Check if the archive_node stores the same expression as another archive_node.
- bool has_ex () const
- ex get_ex () const
- void forget ()

  Delete cached unarchived expressions from node (for debugging).
- void printraw (std::ostream &os) const

  Output archive_node to stream in ugly raw format (for debugging).

Private Attributes

- archive & a

  Reference to the archive to which this node belongs.
- std::vector< property > props

  Vector of stored properties.
- bool has_expression

  Flag indicating whether a cached unarchived representation of this node exists.
- ex e

  The cached unarchived representation of this node (if any).

Friends

- std::ostream & operator<< (std::ostream &os, const archive_node &ar)

  Write archive_node to binary data stream.
- std::istream & operator>> (std::istream &is, archive_node &ar)

  Read archive_node from binary data stream.

8.5.1 Detailed Description

This class stores all properties needed to record/retrieve the state of one object of class basic (or a derived class).

Each property is addressed by its name and data type.

8.5.2 Member Typedef Documentation

8.5.2.1 propinfovector

```
typedef std::vector<property_info> GiNaC::archive_node::propinfovector
```

8.5.2.2 archive_node_cit

```
typedef std::vector<property>::const_iterator GiNaC::archive_node::archive_node_cit
```

8.5.3 Member Enumeration Documentation

8.5.3.1 property_type

```
enum GiNaC::archive_node::property_type
```

Property data types.

Enumerator

PTYPE_BOOL	
PTYPE_UNSIGNED	
PTYPE_STRING	
PTYPE_NODE	

8.5.4 Constructor & Destructor Documentation

8.5.4.1 archive_node() [1/2]

```
GiNaC::archive_node::archive_node (
            archive & ar)  [inline]
```

References a, and has_expression.

Referenced by add_ex(), GiNaC::archive::archive_ex(), find_ex_node(), has_same_ex_as(), operator<<, operator=(), and operator>>.

8.5.4.2 archive_node() [2/2]

```
GiNaC::archive_node::archive_node (
            archive & ar,
            const ex & expr)
```

Recursively construct archive node from expression.

References a, GiNaC::ex::bp, e, and has_expression.

8.5.5 Member Function Documentation

8.5.5.1 operator=()

```
const archive_node & GiNaC::archive_node::operator= (
            const archive_node & other)
```

Assignment operator of archive_node.

References archive_node(), e, has_expression, and props.

8.5.5.2 add_bool()

```
void GiNaC::archive_node::add_bool (
            const std::string & name,
            bool value)
```

Add property of type "bool" to node.

References a, GiNaC::archive::atomize(), GiNaC::archive_node::property::property(), props, and PTYPE_BOOL.

Referenced by GiNaC::minkmetric::archive(), GiNaC::spinidx::archive(), GiNaC::tensepsilon::archive(), and GiNaC::varidx::archive().

8.5.5.3 add_unsigned()

```
void GiNaC::archive_node::add_unsigned (
            const std::string & name,
            unsigned value)
```

Add property of type "unsigned int" to node.

References a, GiNaC::archive::atomize(), GiNaC::archive_node::property::property(), props, and PTYPE_UNSIGNED.

Referenced by GiNaC::clifford::archive(), GiNaC::color::archive(), GiNaC::fderivative::archive(), GiNaC::matrix::archive(), GiNaC::relational::archive(), GiNaC::symmetry::archive(), and GiNaC::wildcard::archive().

8.5.5.4 add_string()

```
void GiNaC::archive_node::add_string (
            const std::string & name,
            const std::string & value)
```

Add property of type "string" to node.

References a, GiNaC::archive::atomize(), GiNaC::archive_node::property::property(), props, and PTYPE_STRING.

Referenced by GiNaC::basic::archive(), GiNaC::constant::archive(), GiNaC::function::archive(), GiNaC::numeric::archive(), and GiNaC::symbol::archive().

8.5.5.5 add_ex()

```
void GiNaC::archive_node::add_ex (
            const std::string & name,
            const ex & value)
```

Add property of type "ex" to node.

References a, GiNaC::archive::add_node(), archive_node(), GiNaC::archive::atomize(), GiNaC::archive_node::property::property(), props, and PTYPE_NODE.

Referenced by GiNaC::clifford::archive(), GiNaC::container< std::vector >::archive(), GiNaC::expairseq::archive(), GiNaC::Gt::archive(), GiNaC::idx::archive(), GiNaC::indexed::archive(), GiNaC::integral::archive(), GiNaC::matrix::archive(), GiNaC::power::archive(), GiNaC::pseries::archive(), GiNaC::relational::archive(), and GiNaC::symmetry::archive().

8.5.5.6 find_bool()

```
bool GiNaC::archive_node::find_bool (
            const std::string & name,
            bool & ret,
            unsigned index = 0) const
```

Retrieve property of type "bool" from node.

Returns

"true" if property was found, "false" otherwise

References a, GiNaC::archive::atomize(), props, and PTYPE_BOOL.

Referenced by GiNaC::minkmetric::read_archive(), GiNaC::spinidx::read_archive(), GiNaC::tensepsilon::read_archive(), and GiNaC::varidx::read_archive().

8.5.5.7 find_unsigned()

```
bool GiNaC::archive_node::find_unsigned (
            const std::string & name,
            unsigned & ret,
            unsigned index = 0) const
```

Retrieve property of type "unsigned" from node.

Returns

"true" if property was found, "false" otherwise

References a, GiNaC::archive::atomize(), props, and PTYPE_UNSIGNED.

Referenced by GiNaC::clifford::read_archive(), GiNaC::color::read_archive(), GiNaC::fderivative::read_archive(), GiNaC::indexed::read_archive(), GiNaC::matrix::read_archive(), GiNaC::relational::read_archive(), GiNaC::symmetry::read_archive(), and GiNaC::wildcard::read_archive().

8.5.5.8 find_string()

```
bool GiNaC::archive_node::find_string (
            const std::string & name,
            std::string & ret,
            unsigned index = 0) const
```

Retrieve property of type "string" from node.

Returns

"true" if property was found, "false" otherwise

References a, GiNaC::archive::atomize(), props, PTYPE_STRING, and GiNaC::archive::unatomize().

Referenced by GiNaC::constant::read_archive(), GiNaC::function::read_archive(), GiNaC::numeric::read_archive(), GiNaC::symbol::read_archive(), and unarchive().

8.5.5.9 find_first()

```
archive_node::archive_node_cit GiNaC::archive_node::find_first (
            const std::string & name) const
```

Find the location in the vector of properties of the first/last property with a given name.

References a, GiNaC::archive::atomize(), and props.

8.5.5.10 find_last()

```
archive_node::archive_node_cit GiNaC::archive_node::find_last (
            const std::string & name) const
```

References a, GiNaC::archive::atomize(), and props.

8.5.5.11 find_property_range()

```
archive_node::archive_node_cit_range GiNaC::archive_node::find_property_range (
            const std::string & name1,
            const std::string & name2) const
```

Find a range of locations in the vector of properties.

The result begins at the first property with name1 and ends one past the last property with name2.

References a, GiNaC::archive::atomize(), GiNaC::archive_node::archive_node_cit_range::begin, GiNaC::archive_node::archive_node and props.

Referenced by GiNaC::container< std::vector >::read_archive(), GiNaC::expairseq::read_archive(), GiNaC::matrix::read_archive(), and GiNaC::pseries::read_archive().

8.5.5.12 find_ex()

```
bool GiNaC::archive_node::find_ex (
            const std::string & name,
            ex & ret,
            lst & sym_lst,
            unsigned index = 0) const
```

Retrieve property of type "ex" from node.

Returns

"true" if property was found, "false" otherwise

References a, GiNaC::archive::atomize(), GiNaC::archive::get_node(), props, PTYPE_NODE, and unarchive().

Referenced by GiNaC::clifford::read_archive(), GiNaC::expairseq::read_archive(), GiNaC::Gt::read_archive(), GiNaC::idx::read_archive(), GiNaC::indexed::read_archive(), GiNaC::integral::read_archive(), GiNaC::power::read_archive(), GiNaC::pseries::read_archive(), GiNaC::relational::read_archive(), and GiNaC::symmetry::read_archive().

8.5.5.13 find_ex_by_loc()

```
void GiNaC::archive_node::find_ex_by_loc (
             archive_node_cit loc,
             ex & ret,
             lst & sym_lst) const
```

Retrieve property of type "ex" from the node if it is known that this node in fact contains such a property at the given location.

This is much more efficient than the preceding function.

References a, GiNaC::archive::get_node(), and unarchive().

Referenced by GiNaC::container< std::vector >::read_archive(), GiNaC::expairseq::read_archive(), GiNaC::matrix::read_archive(), and GiNaC::pseries::read_archive().

8.5.5.14 find_ex_node()

```
const archive_node & GiNaC::archive_node::find_ex_node (
             const std::string & name,
             unsigned index = 0) const
```

Retrieve property of type "ex" from node, returning the node of the sub-expression.

References a, archive_node(), GiNaC::archive::atomize(), GiNaC::archive::get_node(), props, and PTYPE_NODE.

8.5.5.15 get_properties()

```
void GiNaC::archive_node::get_properties (
             propinfovector & v) const
```

Return vector of properties stored in node.

References a, GiNaC::archive_node::property_info::property_info(), props, and GiNaC::archive::unatomize().

8.5.5.16 unarchive()

```
ex GiNaC::archive_node::unarchive (
             lst & sym_lst) const
```

Convert archive node to GiNaC expression.

References GiNaC::status_flags::dynallocated, e, GiNaC::ex::ex(), GiNaC::find_factory_fcn(), find_string(), and has_expression.

Referenced by find_ex(), and find_ex_by_loc().

8.5.5.17 has_same_ex_as()

```
bool GiNaC::archive_node::has_same_ex_as (
            const archive_node & other) const
```

Check if the archive_node stores the same expression as another archive_node.

Returns

"true" if expressions are the same

References archive_node(), GiNaC::ex::bp, e, and has_expression.

8.5.5.18 has_ex()

```
bool GiNaC::archive_node::has_ex () const  [inline]
```

References has_expression.

Referenced by GiNaC::archive::add_node().

8.5.5.19 get_ex()

```
ex GiNaC::archive_node::get_ex () const  [inline]
```

References e.

Referenced by GiNaC::archive::add_node().

8.5.5.20 forget()

```
void GiNaC::archive_node::forget ()
```

Delete cached unarchived expressions from node (for debugging).

References e, and has_expression.

Referenced by GiNaC::archive::forget().

8.5.5.21 printraw()

```
void GiNaC::archive_node::printraw (
            std::ostream & os) const
```

Output archive_node to stream in ugly raw format (for debugging).

References a, GiNaC::ex::bp, e, has_expression, props, PTYPE_BOOL, PTYPE_NODE, PTYPE_STRING, PTYPE_UNSIGNED, and GiNaC::archive::unatomize().

8.5.6 Friends And Related Symbol Documentation

8.5.6.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const archive_node & ar)  [friend]
```

Write archive_node to binary data stream.

References archive_node(), operator<<, props, and GiNaC::write_unsigned().

Referenced by operator<<.

8.5.6.2 operator>>

```
std::istream & operator>> (
            std::istream & is,
            archive_node & ar)  [friend]
```

Read archive_node from binary data stream.

References archive_node(), operator>>, props, and GiNaC::read_unsigned().

Referenced by operator>>.

8.5.7 Member Data Documentation

8.5.7.1 a

```
archive& GiNaC::archive_node::a  [private]
```

Reference to the archive to which this node belongs.

Referenced by add_bool(), add_ex(), add_string(), add_unsigned(), archive_node(), archive_node(), find_bool(), find_ex(), find_ex_by_loc(), find_ex_node(), find_first(), find_last(), find_property_range(), find_string(), find_unsigned(), get_properties(), and printraw().

8.5.7.2 props

```
std::vector<property> GiNaC::archive_node::props  [private]
```

Vector of stored properties.

Referenced by add_bool(), add_ex(), add_string(), add_unsigned(), find_bool(), find_ex(), find_ex_node(), find_first(), find_last(), find_property_range(), find_string(), find_unsigned(), get_properties(), operator<<, operator=(), operator>>, and printraw().

8.5.7.3 has_expression

```
bool GiNaC::archive_node::has_expression  [mutable], [private]
```

Flag indicating whether a cached unarchived representation of this node exists.

Referenced by archive_node(), archive_node(), forget(), has_ex(), has_same_ex_as(), operator=(), printraw(), and unarchive().

8.5.7.4 e

```
ex GiNaC::archive_node::e  [mutable], [private]
```

The cached unarchived representation of this node (if any).

Referenced by archive_node(), forget(), get_ex(), has_same_ex_as(), operator=(), printraw(), and unarchive().

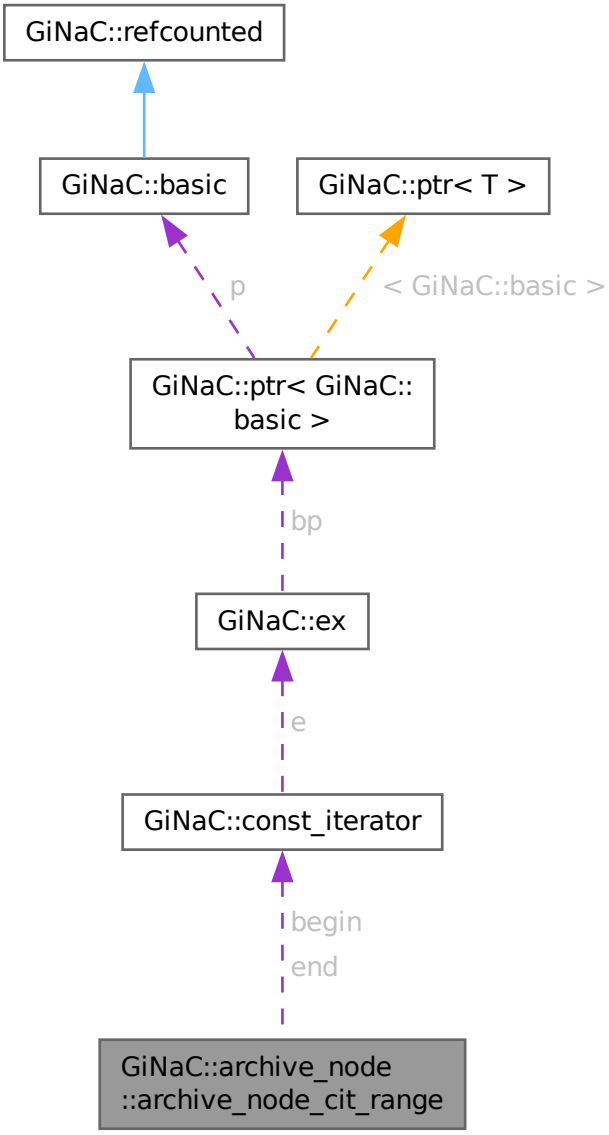
The documentation for this class was generated from the following files:

- archive.h
- archive.cpp

8.6 GiNaC::archive_node::archive_node_cit_range Struct Reference

```
#include <archive.h>
```

Collaboration diagram for GiNaC::archive_node::archive_node_cit_range:



Public Attributes

- archive_node_cit begin
- archive_node_cit end

8.6.1 Member Data Documentation

8.6.1.1 begin

`archive_node_cit GiNaC::archive_node::archive_node_cit_range::begin`

Referenced by GiNaC::archive_node::find_property_range().

8.6.1.2 end

```
archive_node_cit GiNaC::archive_node::archive_node_cit_range::end
```

Referenced by GiNaC::archive_node::find_property_range().

The documentation for this struct was generated from the following file:

- archive.h

8.7 GiNaC::archive::archived_ex Struct Reference

Archived expression descriptor.

Public Member Functions

- archived_ex ()
- archived_ex (archive_atom n, archive_node_id node)

Public Attributes

- archive_atom name
  Name of expression.
- archive_node_id root
  ID of root node.

8.7.1 Detailed Description

Archived expression descriptor.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 archived_ex() [1/2]

```
GiNaC::archive::archived_ex::archived_ex () [inline]
```

Referenced by GiNaC::archive::archive_ex().

8.7.2.2 archived_ex() [2/2]

```
GiNaC::archive::archived_ex::archived_ex (
            archive_atom n,
            archive_node_id node) [inline]
```

References name, and root.

8.7.3 Member Data Documentation

8.7.3.1 name

`archive_atom GiNaC::archive::archived_ex::name`

Name of expression.

Referenced by archived_ex().

8.7.3.2 root

`archive_node_id GiNaC::archive::archived_ex::root`

ID of root node.

Referenced by archived_ex().

The documentation for this struct was generated from the following file:

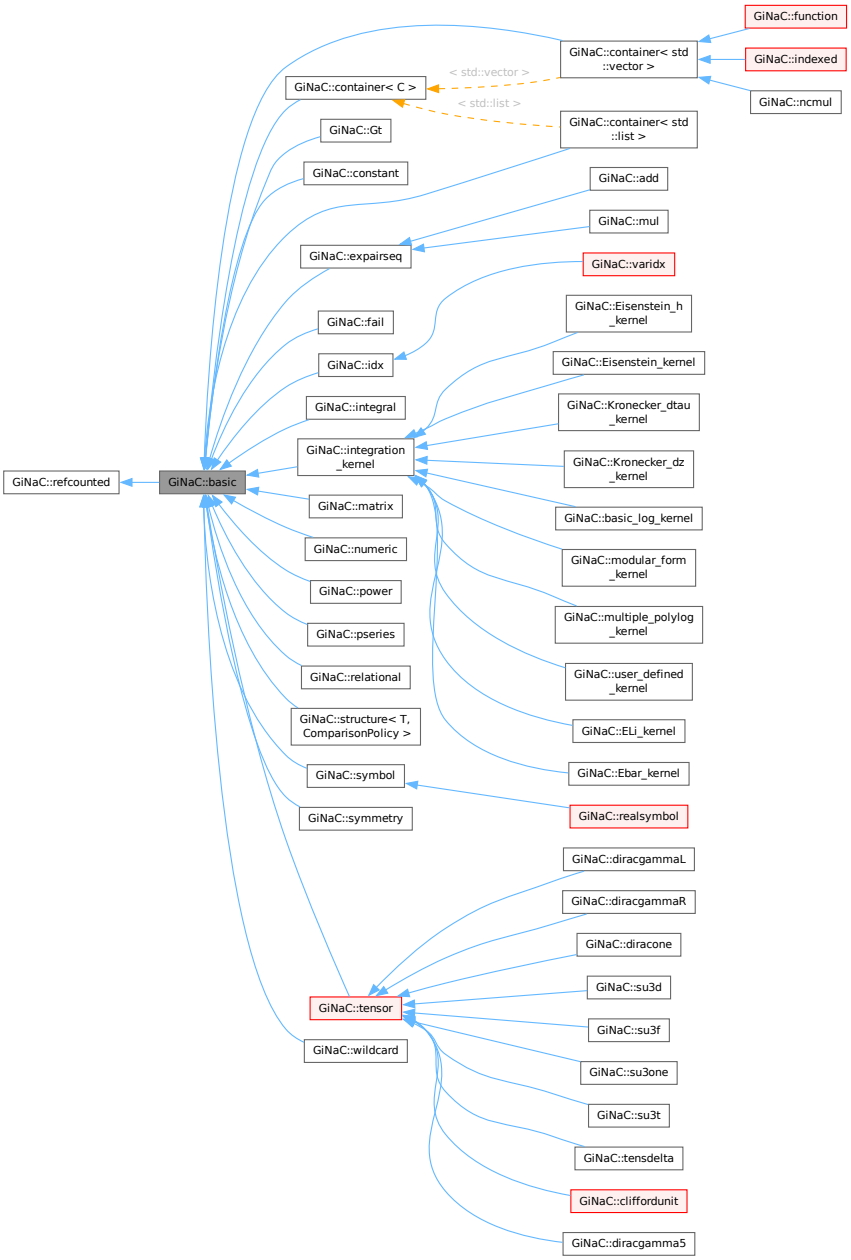
- archive.h

8.8 GiNaC::basic Class Reference

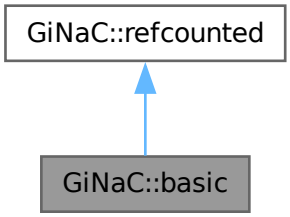
This class is the ABC (abstract base class) of GiNaC's class hierarchy.

```
#include <basic.h>
```

Inheritance diagram for GiNaC::basic:



Collaboration diagram for GiNaC::basic:



Public Member Functions

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const

- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const

- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- unsigned flags
  of type status_flags
- unsigned hashvalue
  hash value

Friends

- class ex

8.8.1 Detailed Description

This class is the ABC (abstract base class) of GiNaC's class hierarchy.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 basic() [1/2]

```
GiNaC::basic::basic ()  [inline], [protected]
```

References basic(), and flags.

Referenced by basic(), basic(), clearflag(), compare(), compare_same_type(), duplicate(), eval_indexed(), GiNaC::matrix::eval_indexed(), GiNaC::structure< T, ComparisonPolicy >::eval_indexed(), hold(), GiNaC::idx::is_dummy_pair_same_
is_equal(), is_equal_same_type(), GiNaC::constant::is_equal_same_type(), GiNaC::container< C >::is_equal_same_type(), GiNaC::container< std::vector >::is_equal_same_type(), GiNaC::expairseq::is_equal_same_type(), GiNaC::numeric::is_equal_same
GiNaC::structure< T, ComparisonPolicy >::is_equal_same_type(), GiNaC::symbol::is_equal_same_type(), map(), match_same_type(), GiNaC::idx::match_same_type(), GiNaC::matrix::match_same_type(), GiNaC::relational::match_same_type(), GiNaC::structure< T, ComparisonPolicy >::match_same_type(), operator=(), setflag(), subs(), and GiNaC::symmetry::sy_swap.

8.8.2.2 ~basic()

```
virtual GiNaC::basic::~basic ()  [inline], [virtual]
```

basic destructor, virtual because class ex will delete objects of derived classes via a basic*.

References GiNaC::status_flags::dynallocated, flags, GiNaC::refcounted::get_refcount(), and GINAC_ASSERT.

8.8.2.3 basic() [2/2]

```
GiNaC::basic::basic (
            const basic & other)
```

References basic(), and operator=().

8.8.3 Member Function Documentation

8.8.3.1 operator=()

```
const basic & GiNaC::basic::operator= (
            const basic & other)
```

basic assignment operator: the other object might be of a derived class.

References basic(), GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, flags, GiNaC::status_flags::hash_calculated, hashvalue, operator=(), and GiNaC::refcounted::set_refcount().

Referenced by basic(), and operator=().

8.8.3.2 duplicate()

```
virtual basic * GiNaC::basic::duplicate () const  [inline], [virtual]
```

Create a clone of this object on the heap.

One can think of this as simulating a virtual copy constructor which is needed for instance by the refcounted construction of an ex from a basic.

Reimplemented in GiNaC::possymbol, and GiNaC::realsymbol.

References basic(), GiNaC::status_flags::dynallocated, and setflag().

Referenced by GiNaC::ex::construct_from_basic(), map(), GiNaC::idx::map(), GiNaC::idx::replace_dim(), subs(), GiNaC::idx::subs(), GiNaC::spinidx::toggle_dot(), GiNaC::varidx::toggle_variance(), and GiNaC::spinidx::toggle_variance_dot().

8.8.3.3 eval()

```
ex GiNaC::basic::eval () const  [virtual]
```

Perform automatic non-interruptive term rewriting rules.

Reimplemented in GiNaC::add, GiNaC::expairseq, GiNaC::fderivative, GiNaC::function, GiNaC::Gt, GiNaC::indexed, GiNaC::integral, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::structure< T, ComparisonPolic and GiNaC::symbol.

References eval(), ex, and hold().

Referenced by GiNaC::ex::construct_from_basic(), and eval().

8.8.3.4 evalf()

```
ex GiNaC::basic::evalf () const  [virtual]
```

Evaluate object numerically.

Reimplemented in GiNaC::constant, GiNaC::function, GiNaC::Gt, GiNaC::idx, GiNaC::integral, GiNaC::mul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, and GiNaC::symbol.

References evalf(), ex, map(), and nops().

Referenced by evalf(), GiNaC::evalf(), GiNaC::Kronecker_dtau_kernel::get_numerical_value(), GiNaC::Kronecker_dtau_kernel::series and GiNaC::Kronecker_dz_kernel::series_coeff_impl().

8.8.3.5 evalm()

```
ex GiNaC::basic::evalm () const  [virtual]
```

Evaluate sums, products and integer powers of matrices.

Reimplemented in GiNaC::add, GiNaC::matrix, GiNaC::mul, GiNaC::ncmul, GiNaC::power, GiNaC::pseries, and GiNaC::structure< T, ComparisonPolicy >.

References evalm(), ex, map(), GiNaC::map_evalm, and nops().

Referenced by evalm().

8.8.3.6 eval_integ()

```
ex GiNaC::basic::eval_integ () const  [virtual]
```

Evaluate integrals, if result is known.

Reimplemented in GiNaC::integral, and GiNaC::pseries.

References eval_integ(), ex, map(), GiNaC::map_eval_integ, and nops().

Referenced by eval_integ().

8.8.3.7 eval_ncmul()

```
ex GiNaC::basic::eval_ncmul (
            const exvector & v) const  [protected], [virtual]
```

Reimplemented in GiNaC::add, GiNaC::clifford, GiNaC::color, GiNaC::function, GiNaC::integral, GiNaC::mul, GiNaC::power, GiNaC::relational, and GiNaC::structure< T, ComparisonPolicy >.

References eval_ncmul(), ex, and GiNaC::hold_ncmul().

Referenced by eval_ncmul().

8.8.3.8 eval_indexed()

```
ex GiNaC::basic::eval_indexed (
            const basic & i) const  [virtual]
```

Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

Reimplemented in GiNaC::matrix, GiNaC::minkmetric, GiNaC::spinmetric, GiNaC::structure< T, ComparisonPolicy >, GiNaC::su3d, GiNaC::su3f, GiNaC::tensdelta, GiNaC::tensepsilon, and GiNaC::tensmetric.

References basic(), eval_indexed(), and ex.

Referenced by eval_indexed().

8.8.3.9 print()

```
void GiNaC::basic::print (
             const print_context & c,
             unsigned level = 0) const  [virtual]
```

Output to stream.

This performs double dispatch on the dynamic type of ∗this and the dynamic type of the supplied print context.

Parameters

c	print context object that describes the output formatting
level	value that is used to identify the precedence or indentation level for placing parentheses and formatting

Reimplemented in GiNaC::fderivative, GiNaC::function, and GiNaC::structure< T, ComparisonPolicy >.

References print(), and print_dispatch().

Referenced by dbgprint(), dbgprinttree(), print(), GiNaC::fderivative::print(), GiNaC::add::print_add(), GiNaC::mul::print_overall_coeff(), GiNaC::pseries::print_series(), and GiNaC::print_sym_pow().

8.8.3.10 dbgprint()

```
void GiNaC::basic::dbgprint () const  [virtual]
```

Little wrapper around print to be called within a debugger.

This is needed because you cannot call foo.print(cout) from within the debugger because it might not know what cout is. This method can be invoked with no argument and it will simply print to stdout.

See also

basic::print

basic::dbgprinttree

References dbgprint(), and print().

Referenced by dbgprint().

8.8.3.11 dbgprinttree()

```
void GiNaC::basic::dbgprinttree () const  [virtual]
```

Little wrapper around printtree to be called within a debugger.

See also

basic::dbgprint

References dbgprinttree(), and print().

Referenced by dbgprinttree().

8.8.3.12 precedence()

```
unsigned GiNaC::basic::precedence () const  [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented in GiNaC::add, GiNaC::clifford, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::expairseq, GiNaC::function, GiNaC::Gt, GiNaC::indexed, GiNaC::integral, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::relational, and GiNaC::structure< T, Compariso

References precedence().

Referenced by precedence().

8.8.3.13 info()

```
bool GiNaC::basic::info (
            unsigned inf) const  [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented in GiNaC::add, GiNaC::constant, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::expairseq, GiNaC::function, GiNaC::idx, GiNaC::indexed, GiNaC::minkmetric, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::relational, GiNaC::spinmetric, GiNaC::structure< T, Comparison GiNaC::symbol, GiNaC::tensdelta, GiNaC::tensepsilon, and GiNaC::tensmetric.

References info().

Referenced by info(), GiNaC::function::info(), and GiNaC::matrix::solve().

8.8.3.14 nops()

```
size_t GiNaC::basic::nops () const  [virtual]
```

Number of operands/members.

Reimplemented in GiNaC::clifford, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::Ebar_kernel, GiNaC::Eisenstein_h_kernel, GiNaC::Eisenstein_kernel, GiNaC::ELi_kernel, GiNaC::expairseq, GiNaC::Gt, GiNaC::idx, GiNaC::integral, GiNaC::Kronecker_dtau_kernel, GiNaC::Kronecker_dz_kernel, GiNaC::matrix, GiNaC::modular_form_kernel, GiNaC::multiple_polylog_kernel, GiNaC::power, GiNaC::pseries, GiNaC::relational, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::user_defined_kernel.

References nops().

Referenced by calchash(), derivative(), do_print_tree(), GiNaC::matrix::eval_indexed(), GiNaC::minkmetric::eval_indexed(), GiNaC::spinmetric::eval_indexed(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::tensepsilon::eval_indexed(), GiNaC::tensmetric::eval_indexed(), eval_integ(), evalf(), evalm(), expand(), has(), map(), match(), nops(), normal(), and subs().

8.8.3.15 op()

```
ex GiNaC::basic::op (
            size_t i) const  [virtual]
```

Return operand/member at position i.

Reimplemented in GiNaC::clifford, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::Ebar_kernel, GiNaC::Eisenstein_h_kernel, GiNaC::Eisenstein_kernel, GiNaC::ELi_kernel, GiNaC::expairseq, GiNaC::idx, GiNaC::integral, GiNaC::Kronecker_dtau_kernel, GiNaC::Kronecker_dz_kernel, GiNaC::matrix, GiNaC::modular_form_kernel, GiNaC::multiple_polylog_kernel, GiNaC::power, GiNaC::pseries, GiNaC::relational, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::user_defined_kernel.

References ex, and op().

Referenced by calchash(), do_print_tree(), GiNaC::matrix::eval_indexed(), GiNaC::minkmetric::eval_indexed(), GiNaC::spinmetric::eval_indexed(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::tensepsilon::eval_indexed(), GiNaC::tensmetric::eval_indexed(), has(), map(), match(), op(), operator[](), operator[](), and subs().

8.8.3.16 operator[]() [1/4]

```
ex GiNaC::basic::operator[] (
            const ex & index) const  [virtual]
```

Reimplemented in GiNaC::structure< T, ComparisonPolicy >.

References ex, GiNaC::ex_to(), GiNaC::is_exactly_a(), op(), operator[](), and GiNaC::to_int().

Referenced by operator[](), operator[](), operator[](), and operator[]().

8.8.3.17 operator[]() [2/4]

```
ex GiNaC::basic::operator[] (
            size_t i) const  [virtual]
```

Reimplemented in GiNaC::structure< T, ComparisonPolicy >.

References ex, op(), and operator[]().

8.8.3.18 let_op()

```
ex & GiNaC::basic::let_op (
            size_t i)  [virtual]
```

Return modifiable operand/member at position i.

Reimplemented in GiNaC::clifford, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::Ebar_kernel, GiNaC::Eisenstein_h_kernel, GiNaC::Eisenstein_kernel, GiNaC::ELi_kernel, GiNaC::integral, GiNaC::Kronecker_dtau_kernel, GiNaC::Kronecker_dz_kernel, GiNaC::matrix, GiNaC::modular_form_kernel, GiNaC::multiple_polylog_kernel, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::user_defined_kernel.

References ensure_if_modifiable(), ex, and let_op().

Referenced by let_op(), map(), operator[](), operator[](), and subs().

8.8.3.19 operator[]() [3/4]

```
ex & GiNaC::basic::operator[] (
             const ex & index)  [virtual]
```

Reimplemented in GiNaC::structure< T, ComparisonPolicy >.

References ex, GiNaC::ex_to(), GiNaC::is_exactly_a(), let_op(), operator[](), and GiNaC::to_int().

8.8.3.20 operator[]() [4/4]

```
ex & GiNaC::basic::operator[] (
             size_t i)  [virtual]
```

Reimplemented in GiNaC::structure< T, ComparisonPolicy >.

References ex, let_op(), and operator[]().

8.8.3.21 has()

```
bool GiNaC::basic::has (
             const ex & pattern,
             unsigned options = 0) const  [virtual]
```

Test for occurrence of a pattern.

An object 'has' a pattern if it matches the pattern itself or one of the children 'has' it. As a consequence (according to the definition of children) given e=x+y+z, e.has(x) is true but e.has(x+y) is false.

Reimplemented in GiNaC::Gt, GiNaC::mul, GiNaC::numeric, GiNaC::power, and GiNaC::structure< T, ComparisonPolicy >.

References ex, has(), match(), nops(), and op().

Referenced by has(), GiNaC::mul::has(), GiNaC::power::has(), is_polynomial(), and series().

8.8.3.22 match()

```
bool GiNaC::basic::match (
             const ex & pattern,
             exmap & repl_lst) const  [virtual]
```

Check whether the expression matches a given pattern.

For every wildcard object in the pattern, a pair with the wildcard as a key and matching expression as a value is added to repl_lst.

Reimplemented in GiNaC::expairseq, GiNaC::Gt, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::wildcard.

References ex, GiNaC::ex_to(), is_equal(), is_equal_same_type(), GiNaC::is_exactly_a(), match(), match_same_type(), nops(), GiNaC::ex::nops(), op(), and GiNaC::ex::op().

Referenced by has(), match(), and subs_one_level().

8.8.3.23 match_same_type()

```
bool GiNaC::basic::match_same_type (
            const basic & other) const  [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented in GiNaC::clifford, GiNaC::color, GiNaC::fderivative, GiNaC::function, GiNaC::idx, GiNaC::matrix, GiNaC::relational, GiNaC::spinidx, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::varidx.

References basic(), and match_same_type().

Referenced by match(), and match_same_type().

8.8.3.24 subs()

```
ex GiNaC::basic::subs (
            const exmap & m,
            unsigned options = 0) const  [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented in GiNaC::clifford, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::expairseq, GiNaC::Gt, GiNaC::idx, GiNaC::matrix, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::relational, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::symbol.

References GiNaC::are_ex_trivially_equal(), basic(), clearflag(), duplicate(), ex, GiNaC::status_flags::expanded, GiNaC::status_flags::hash_calculated, let_op(), nops(), op(), subs(), GiNaC::ex::subs(), and subs_one_level().

Referenced by GiNaC::tensmetric::eval_indexed(), and subs().

8.8.3.25 map()

```
ex GiNaC::basic::map (
            map_function & f) const  [virtual]
```

Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

Reimplemented in GiNaC::expairseq, GiNaC::idx, GiNaC::power, GiNaC::relational, and GiNaC::structure< T, ComparisonPolicy >.

References GiNaC::are_ex_trivially_equal(), basic(), clearflag(), duplicate(), ex, GiNaC::status_flags::expanded, GiNaC::status_flags::hash_calculated, let_op(), map(), nops(), and op().

Referenced by derivative(), eval_integ(), evalf(), evalm(), expand(), map(), and normal().

8.8.3.26 accept()

```
virtual void GiNaC::basic::accept (
            GiNaC::visitor & v) const  [inline], [virtual]
```

8.8.3.27 is_polynomial()

```
bool GiNaC::basic::is_polynomial (
            const ex & var) const  [virtual]
```

Check whether this is a polynomial in the given variables.

Reimplemented in GiNaC::add, GiNaC::constant, GiNaC::mul, GiNaC::numeric, GiNaC::power, and GiNaC::symbol.

References ex, GiNaC::ex_to(), has(), is_equal(), and is_polynomial().

Referenced by is_polynomial().

8.8.3.28 degree()

```
int GiNaC::basic::degree (
            const ex & s) const  [virtual]
```

Return degree of highest power in object s.

Reimplemented in GiNaC::add, GiNaC::integral, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, and GiNaC::structure< T, ComparisonPolicy >.

References degree(), ex, GiNaC::ex_to(), and is_equal().

Referenced by collect(), and degree().

8.8.3.29 ldegree()

```
int GiNaC::basic::ldegree (
            const ex & s) const  [virtual]
```

Return degree of lowest power in object s.

Reimplemented in GiNaC::add, GiNaC::integral, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, and GiNaC::structure< T, ComparisonPolicy >.

References ex, GiNaC::ex_to(), is_equal(), and ldegree().

Referenced by collect(), and ldegree().

8.8.3.30 coeff()

```
ex GiNaC::basic::coeff (
            const ex & s,
            int n = 1) const  [virtual]
```

Return coefficient of degree n in object s.

Reimplemented in GiNaC::add, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, and GiNaC::structure< T, ComparisonPolicy >.

References GiNaC::_ex0, GiNaC::_ex1, coeff(), ex, GiNaC::ex_to(), and is_equal().

Referenced by coeff(), collect(), GiNaC::expairseq::read_archive(), series(), GiNaC::indexed::simplify_indexed, and GiNaC::sqrfree_parfrac().

8.8.3.31 expand()

```
ex GiNaC::basic::expand (
            unsigned options = 0) const  [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented in GiNaC::add, GiNaC::expairseq, GiNaC::function, GiNaC::indexed, GiNaC::integral, GiNaC::mul, GiNaC::ncmul, GiNaC::power, GiNaC::pseries, and GiNaC::structure< T, ComparisonPolicy >.

References ex, GiNaC::ex_to(), expand(), GiNaC::status_flags::expanded, map(), nops(), and setflag().

Referenced by collect(), expand(), GiNaC::expand(), GiNaC::matrix::fraction_free_elimination(), and GiNaC::matrix::pivot().

8.8.3.32 collect()

```
ex GiNaC::basic::collect (
            const ex & s,
            bool distributed = false) const  [virtual]
```

Sort expanded expression in terms of powers of some object(s).

Parameters

s	object(s) to sort in
distributed	recursive or distributed form (only used when s is a list)

Reimplemented in GiNaC::pseries, and GiNaC::structure< T, ComparisonPolicy >.

References GiNaC::_ex0, GiNaC::_ex1, coeff(), GiNaC::ex::coeff(), collect(), GiNaC::ex::collect(), degree(), GiNaC::ex::degree(), GiNaC::dynallocate(), ex, GiNaC::ex_to(), expand(), GiNaC::ex::find(), GiNaC::is_a(), ldegree(), GiNaC::ex::nops(), GiNaC::ex::op(), and GiNaC::pow().

Referenced by collect().

8.8.3.33 derivative()

```
ex GiNaC::basic::derivative (
            const symbol & s) const  [protected], [virtual]
```

Default implementation of ex::diff().

It maps the operation on the operands (or returns 0 when the object has no operands).

See also

ex::diff

Reimplemented in GiNaC::add, GiNaC::constant, GiNaC::fderivative, GiNaC::function, GiNaC::idx, GiNaC::indexed, GiNaC::integral, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::structure< T, ComparisonPolic and GiNaC::symbol.

References GiNaC::_ex0, derivative(), ex, map(), and nops().

Referenced by derivative(), and diff().

8.8.3.34 series()

```
ex GiNaC::basic::series (
            const relational & r,
            int order,
            unsigned options = 0) const  [virtual]
```

Default implementation of ex::series().

This performs Taylor expansion.

See also

ex::series

Reimplemented in GiNaC::add, GiNaC::Eisenstein_h_kernel, GiNaC::Eisenstein_kernel, GiNaC::fderivative, GiNaC::function, GiNaC::integral, GiNaC::integration_kernel, GiNaC::modular_form_kernel, GiNaC::mul, GiNaC::power, GiNaC::pseries, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::symbol.

References GiNaC::_ex0, GiNaC::_ex1, coeff(), GiNaC::ex::diff(), GiNaC::numeric::div(), ex, GiNaC::ex_to(), GiNaC::expair::expair(), GiNaC::ex::expand(), has(), GiNaC::ex::is_zero(), GiNaC::relational::lhs(), GiNaC::subs_options::no_pattern, GiNaC::pseries::pseries(), and GiNaC::ex::subs().

Referenced by GiNaC::lgamma_series(), GiNaC::fderivative::series(), GiNaC::function::series(), and GiNaC::power::series().

8.8.3.35 normal()

```
ex GiNaC::basic::normal (
             exmap & repl,
             exmap & rev_lookup,
             lst & modifier) const  [virtual]
```

Default implementation of ex::normal().

It normalizes the children and replaces the object with a temporary symbol.

See also

ex::normal

Reimplemented in GiNaC::add, GiNaC::mul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::structure< T, ComparisonPolic and GiNaC::symbol.

References GiNaC::_ex1, GiNaC::dynallocate(), ex, GiNaC::ex::info(), GiNaC::is_a(), map(), GiNaC::info_flags::negative, GiNaC::subs_options::no_pattern, nops(), GiNaC::container< C >::nops(), GiNaC::container< C >::op(), GiNaC::ex::op(), GiNaC::power::power(), GiNaC::replace_with_symbol(), and GiNaC::ex::subs().

Referenced by GiNaC::matrix::gauss_elimination(), GiNaC::matrix::markowitz_elimination(), GiNaC::matrix::solve(), and GiNaC::Gt::tauToFundamental().

8.8.3.36 to_rational()

```
ex GiNaC::basic::to_rational (
             exmap & repl) const  [virtual]
```

Default implementation of ex::to_rational().

This replaces the object with a temporary symbol.

Reimplemented in GiNaC::expairseq, GiNaC::numeric, GiNaC::power, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::symbol.

References ex, and GiNaC::replace_with_symbol().

8.8.3.37 to_polynomial()

```
ex GiNaC::basic::to_polynomial (
             exmap & repl) const  [virtual]
```

Reimplemented in GiNaC::expairseq, GiNaC::numeric, GiNaC::power, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::symbol.

References ex, and GiNaC::replace_with_symbol().

8.8.3.38 integer_content()

```
numeric GiNaC::basic::integer_content () const  [virtual]
```

Reimplemented in GiNaC::add, GiNaC::mul, GiNaC::numeric, and GiNaC::structure< T, ComparisonPolicy >.

References GiNaC::_num1_p.

8.8.3.39 smod()

```
ex GiNaC::basic::smod (
            const numeric & xi) const  [virtual]
```

Apply symmetric modular homomorphism to an expanded multivariate polynomial.

This function is usually used internally by heur_gcd().

Parameters

xi	modulus

Returns

mapped polynomial

See also

heur_gcd

Reimplemented in GiNaC::add, GiNaC::mul, GiNaC::numeric, and GiNaC::structure< T, ComparisonPolicy >.

References ex.

8.8.3.40 max_coefficient()

```
numeric GiNaC::basic::max_coefficient () const  [virtual]
```

Implementation ex::max_coefficient().

See also

heur_gcd

Reimplemented in GiNaC::add, GiNaC::mul, GiNaC::numeric, and GiNaC::structure< T, ComparisonPolicy >.

References GiNaC::_num1_p.

8.8.3.41 get_free_indices()

```
exvector GiNaC::basic::get_free_indices () const  [virtual]
```

Return a vector containing the free indices of an expression.

Reimplemented in GiNaC::add, GiNaC::indexed, GiNaC::integral, GiNaC::mul, GiNaC::ncmul, and GiNaC::structure< T, ComparisonP

References get_free_indices().

Referenced by get_free_indices().

8.8.3.42 add_indexed()

```
ex GiNaC::basic::add_indexed (
            const ex & self,
            const ex & other) const  [virtual]
```

Add two indexed expressions.

They are guaranteed to be of class indexed (or a subclass) and their indices are compatible. This function is used internally by simplify_indexed().

Parameters

self	First indexed expression; its base object is ∗this
other	Second indexed expression

Returns

sum of self and other

See also

ex::simplify_indexed()

Reimplemented in GiNaC::matrix, and GiNaC::structure< T, ComparisonPolicy >.

References add_indexed(), and ex.

Referenced by add_indexed().

8.8.3.43 scalar_mul_indexed()

```
ex GiNaC::basic::scalar_mul_indexed (
             const ex & self,
             const numeric & other) const  [virtual]
```

Multiply an indexed expression with a scalar.

This function is used internally by simplify_indexed().

Parameters

self	Indexed expression; its base object is ∗this
other	Numeric value

Returns

product of self and other

See also

ex::simplify_indexed()

Reimplemented in GiNaC::matrix, and GiNaC::structure< T, ComparisonPolicy >.

References ex, and scalar_mul_indexed().

Referenced by scalar_mul_indexed().

8.8.3.44 contract_with()

```
bool GiNaC::basic::contract_with (
             exvector::iterator self,
             exvector::iterator other,
             exvector & v) const  [virtual]
```

Try to contract two indexed expressions that appear in the same product.

If a contraction exists, the function overwrites one or both of the expressions and returns true. Otherwise it returns false. It is guaranteed that both expressions are of class indexed (or a subclass) and that at least one dummy index has been found. This functions is used internally by simplify_indexed().

Parameters

self	Pointer to first indexed expression; its base object is ∗this
other	Pointer to second indexed expression
v	The complete vector of factors

Returns

true if the contraction was successful, false otherwise

See also

ex::simplify_indexed()

Reimplemented in GiNaC::cliffordunit, GiNaC::diracgamma, GiNaC::matrix, GiNaC::spinmetric, GiNaC::structure< T, ComparisonPoli
GiNaC::su3d, GiNaC::su3f, GiNaC::su3t, GiNaC::tensdelta, GiNaC::tensepsilon, and GiNaC::tensmetric.

References contract_with().

Referenced by contract_with().

8.8.3.45 return_type()

```
unsigned GiNaC::basic::return_type () const  [virtual]
```

Reimplemented in GiNaC::add, GiNaC::clifford, GiNaC::color, GiNaC::expairseq, GiNaC::fail, GiNaC::function, GiNaC::indexed, GiNaC::integral, GiNaC::matrix, GiNaC::minkmetric, GiNaC::mul, GiNaC::ncmul, GiNaC::power, GiNaC::relational, GiNaC::structure< T, ComparisonPolicy >, GiNaC::su3d, GiNaC::su3f, GiNaC::tensdelta, GiNaC::tensepsilon, GiNaC::tensmetric, and GiNaC::tensor.

References GiNaC::return_types::commutative, and return_type().

Referenced by return_type().

8.8.3.46 return_type_tinfo()

```
return_type_t GiNaC::basic::return_type_tinfo () const  [virtual]
```

Reimplemented in GiNaC::add, GiNaC::clifford, GiNaC::color, GiNaC::function, GiNaC::indexed, GiNaC::integral, GiNaC::mul, GiNaC::ncmul, GiNaC::power, GiNaC::relational, and GiNaC::structure< T, ComparisonPolicy >.

References return_type_tinfo(), GiNaC::return_type_t::rl, and GiNaC::return_type_t::tinfo.

Referenced by return_type_tinfo().

8.8.3.47 conjugate()

```
ex GiNaC::basic::conjugate () const  [virtual]
```

Reimplemented in GiNaC::add, GiNaC::constant, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::diracgamma5, GiNaC::diracgammaL, GiNaC::diracgammaR, GiNaC::expairseq, GiNaC::function, GiNaC::integral, GiNaC::matrix, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::realsymbol, GiNaC::spinidx, and GiNaC::symbol.

References conjugate(), and ex.

Referenced by conjugate().

8.8.3.48 real_part()

```
ex GiNaC::basic::real_part () const  [virtual]
```

Reimplemented in GiNaC::add, GiNaC::constant, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::function, GiNaC::indexed, GiNaC::matrix, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::realsymbol, and GiNaC::symbol.

References ex, and real_part().

Referenced by real_part(), GiNaC::function::real_part(), and GiNaC::ncmul::real_part().

8.8.3.49 imag_part()

```
ex GiNaC::basic::imag_part () const  [virtual]
```

Reimplemented in GiNaC::add, GiNaC::constant, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::function, GiNaC::indexed, GiNaC::matrix, GiNaC::mul, GiNaC::ncmul, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::realsymbol, and GiNaC::symbol.

References ex, and imag_part().

Referenced by imag_part(), GiNaC::function::imag_part(), and GiNaC::ncmul::imag_part().

8.8.3.50 compare_same_type()

```
int GiNaC::basic::compare_same_type (
            const basic & other) const  [protected], [virtual]
```

Returns order relation between two objects of same type.

This needs to be implemented by each class. It may never return anything else than 0, signalling equality, or +1 and -1 signalling inequality and determining the canonical ordering. (Perl hackers will wonder why C++ doesn't feature the spaceship operator <=> for denoting just this.)

References basic(), GiNaC::compare_pointers(), and compare_same_type().

Referenced by compare(), compare_same_type(), GiNaC::symbol::derivative(), and is_equal_same_type().

8.8.3.51 is_equal_same_type()

```
bool GiNaC::basic::is_equal_same_type (
            const basic & other) const  [protected], [virtual]
```

Returns true if two objects of same type are equal.

Normally needs not be reimplemented as long as it wasn't overwritten by some parent class, since it just calls compare_same_type(). The reason why this function exists is that sometimes it is easier to determine equality than an order relation and then it can be overridden.

Reimplemented in GiNaC::constant, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::expairseq, GiNaC::fderivative, GiNaC::function, GiNaC::numeric, GiNaC::structure< T, ComparisonPolicy >, and GiNaC::symbol.

References basic(), compare_same_type(), and is_equal_same_type().

Referenced by is_equal(), is_equal_same_type(), and match().

8.8.3.52 calchash()

```
unsigned GiNaC::basic::calchash () const  [protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented in GiNaC::constant, GiNaC::expairseq, GiNaC::function, GiNaC::Gt, GiNaC::idx, GiNaC::numeric, GiNaC::relational, GiNaC::structure< T, ComparisonPolicy >, GiNaC::symbol, GiNaC::symmetry, and GiNaC::wildcard.

References calchash(), GiNaC::status_flags::evaluated, flags, GiNaC::ex::gethash(), GiNaC::status_flags::hash_calculated, hashvalue, nops(), op(), GiNaC::rotate_left(), and setflag().

Referenced by calchash(), and gethash().

8.8.3.53 print_dispatch() [1/2]

```
template<class T>
void GiNaC::basic::print_dispatch (
             const print_context & c,
             unsigned level) const  [inline]
```

Like print(), but dispatch to the specified class.

Can be used by implementations of print methods to dispatch to the method of the superclass.

See also

basic::print

References print_dispatch().

Referenced by GiNaC::clifford::do_print_dflt(), GiNaC::clifford::do_print_latex(), print(), print_dispatch(), and print_dispatch().

8.8.3.54 print_dispatch() [2/2]

```
void GiNaC::basic::print_dispatch (
             const registered_class_info & ri,
             const print_context & c,
             unsigned level) const
```

Like print(), but dispatch to the specified class.

Can be used by implementations of print methods to dispatch to the method of the superclass.

See also

basic::print

References GiNaC::print_context_options::get_id(), GiNaC::class_info< OPT >::get_parent(), GiNaC::registered_class_options::get_
GiNaC::class_info< OPT >::options, print_dispatch(), and GiNaC::structure< T, CP >::reg_info.

8.8.3.55 archive()

```
void GiNaC::basic::archive (
            archive_node & n) const  [virtual]
```

Save (serialize) the object into archive node.

Archive the object.

Losely speaking, this method turns an expression into a byte stream (which can be saved and restored later on, or sent via network, etc.)

Reimplemented in GiNaC::clifford, GiNaC::color, GiNaC::constant, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::expairseq, GiNaC::fderivative, GiNaC::function, GiNaC::Gt, GiNaC::idx, GiNaC::indexed, GiNaC::integral, GiNaC::matrix, GiNaC::minkmetric, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::relational, GiNaC::spinidx, GiNaC::symbol, GiNaC::symmetry, GiNaC::tensepsilon, GiNaC::varidx, and GiNaC::wildcard.

References GiNaC::archive_node::add_string(), and archive().

Referenced by archive().

8.8.3.56 read_archive()

```
void GiNaC::basic::read_archive (
            const archive_node & n,
            lst & syms)  [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented in GiNaC::clifford, GiNaC::color, GiNaC::constant, GiNaC::container< C >, GiNaC::container< std::list >, GiNaC::container< std::vector >, GiNaC::expairseq, GiNaC::fderivative, GiNaC::function, GiNaC::Gt, GiNaC::idx, GiNaC::indexed, GiNaC::integral, GiNaC::matrix, GiNaC::minkmetric, GiNaC::numeric, GiNaC::power, GiNaC::pseries, GiNaC::relational, GiNaC::spinidx, GiNaC::symbol, GiNaC::symmetry, GiNaC::tensepsilon, GiNaC::varidx, and GiNaC::wildcard.

References read_archive().

Referenced by read_archive().

8.8.3.57 subs_one_level()

```
ex GiNaC::basic::subs_one_level (
            const exmap & m,
            unsigned options) const
```

Helper function for subs().

Does not recurse into subexpressions.

References ex, GiNaC::ex_to(), match(), GiNaC::subs_options::no_pattern, and subs_one_level().

Referenced by GiNaC::mul::algebraic_subs_mul(), subs(), GiNaC::container< C >::subs(), GiNaC::expairseq::subs(), GiNaC::numeric::subs(), GiNaC::power::subs(), GiNaC::relational::subs(), GiNaC::symbol::subs(), and subs_one_level().

8.8.3.58 diff()

```
ex GiNaC::basic::diff (
            const symbol & s,
            unsigned nth = 1) const
```

Default interface of nth derivative ex::diff(s, n).

It should be called instead of ::derivative(s) for first derivatives and for nth derivatives it just recurses down.

Parameters

s	symbol to differentiate in
nth	order of differentiation

See also

ex::diff

References derivative(), diff(), GiNaC::ex::diff(), GiNaC::status_flags::evaluated, ex, flags, and GiNaC::ex::is_zero().

Referenced by GiNaC::Gt::decomposeIntegrationPath(), GiNaC::mul::derivative(), and diff().

8.8.3.59 compare()

```
int GiNaC::basic::compare (
            const basic & other) const
```

Compare objects syntactically to establish canonical ordering.

All compare functions return: -1 for ∗this less than other, 0 equal, 1 greater.

References basic(), compare(), compare_same_type(), and gethash().

Referenced by compare().

8.8.3.60 is_equal()

```
bool GiNaC::basic::is_equal (
            const basic & other) const
```

Test for syntactic equality.

This is only a quick test, meaning objects should be in the same domain. You might have to .expand(), .normal() objects first, depending on the domain of your computation, to get a more reliable answer.

See also

is_equal_same_type

References basic(), gethash(), is_equal(), and is_equal_same_type().

Referenced by coeff(), GiNaC::ncmul::coeff(), GiNaC::power::coeff(), degree(), GiNaC::ncmul::degree(), GiNaC::power::degree(), is_equal(), is_polynomial(), ldegree(), GiNaC::ncmul::ldegree(), GiNaC::power::ldegree(), match(), and GiNaC::wildcard::match().

8.8.3.61 hold()

```
const basic & GiNaC::basic::hold () const
```

Stop further evaluation.

See also

basic::eval

References basic(), GiNaC::status_flags::evaluated, hold(), and setflag().

Referenced by GiNaC::abs_conjugate(), GiNaC::abs_eval(), GiNaC::abs_evalf(), GiNaC::abs_expand(), GiNaC::abs_real_part(), GiNaC::acos_eval(), GiNaC::acos_evalf(), GiNaC::acosh_eval(), GiNaC::acosh_evalf(), GiNaC::asin_eval(), GiNaC::asin_evalf(), GiNaC::asinh_eval(), GiNaC::asinh_evalf(), GiNaC::atan_eval(), GiNaC::atan_evalf(), GiNaC::atanh_eval(), GiNaC::atanh_evalf(), GiNaC::binomial_conjugate(), GiNaC::binomial_eval(), GiNaC::binomial_evalf(), GiNaC::binomial_real_part(), GiNaC::binomial_sym(), GiNaC::cos_eval(), GiNaC::cos_evalf(), GiNaC::cosh_eval(), GiNaC::cosh_evalf(), GiNaC::eta_imag_part(), GiNaC::add::eval(), eval(), GiNaC::expairseq::eval(), GiNaC::fderivative::eval(), GiNaC::function::eval(), GiNaC::Gt::eval(), GiNaC::integral::eval(), GiNaC::mul::eval(), GiNaC::numeric::eval(), GiNaC::power::eval(), GiNaC::pseries::eval(), GiNaC::structure< T, ComparisonPolicy >::eval(), GiNaC::matrix::eval_indexed(), GiNaC::spinmetric::eval_indexed(), GiNaC::structure< T, ComparisonPolicy >::eval_indexed(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::tensepsilon::eval_indexed(), GiNaC::tensmetric::eval_indexed(), GiNaC::exp_eval(), GiNaC::exp_evalf(), GiNaC::exp_expand(), GiNaC::power::expand(), GiNaC::factorial_conjugate(), GiNaC::factorial_eval(), GiNaC::factorial_evalf(), GiNaC::factorial_real_part(), GiNaC::G2_eval(), GiNaC::G2_evalf(), GiNaC::G3_eval(), GiNaC::G3_evalf(), GiNaC::Gt_detail::pathintegral_term::G_path(), GiNaC::H_eval(), hold(), GiNaC::iterated_integral2_eval(), GiNaC::iterated_integral3_eval(), GiNaC::iterated_integral_evalf_impl(), GiNaC::Li2_conjugate(), GiNaC::Li2_eval(), GiNaC::Li2_evalf(), GiNaC::log_eval(), GiNaC::log_evalf(), GiNaC::log_expand(), GiNaC::log_real_part(), GiNaC::psi1_eval(), GiNaC::psi1_evalf(), GiNaC::psi2_eval(), GiNaC::psi2_evalf(), GiNaC::sin_eval(), GiNaC::sin_evalf(), GiNaC::sinh_eval(), GiNaC::sinh_evalf(), GiNaC::step_conjugate(), GiNaC::step_eval(), GiNaC::step_evalf(), GiNaC::step_real_part(), GiNaC::tan_eval(), GiNaC::tan_evalf(), GiNaC::tanh_eval(), GiNaC::tanh_evalf(), GiNaC::zeta1_eval(), GiNaC::zeta1_evalf(), GiNaC::zeta2_eval(), GiNaC::zeta2_evalf(), and GiNaC::zetaderiv_eval().

8.8.3.62 gethash()

```
unsigned GiNaC::basic::gethash () const  [inline]
```

References calchash(), flags, GiNaC::status_flags::hash_calculated, and hashvalue.

Referenced by GiNaC::remember_table::add_entry(), GiNaC::Gt::calchash(), compare(), is_equal(), GiNaC::remember_table_entry::is and GiNaC::remember_table::lookup_entry().

8.8.3.63 setflag()

```
const basic & GiNaC::basic::setflag (
            unsigned f) const  [inline]
```

Set some status_flags.

References basic(), and flags.

Referenced by calchash(), GiNaC::constant::calchash(), GiNaC::expairseq::calchash(), GiNaC::function::calchash(), GiNaC::Gt::calchash(), GiNaC::idx::calchash(), GiNaC::numeric::calchash(), GiNaC::relational::calchash(), GiNaC::symbol::calchash(), GiNaC::symmetry::calchash(), GiNaC::wildcard::calchash(), GiNaC::constant::constant(), GiNaC::constant::constant(), GiNaC::ex::construct_from_basic(), GiNaC::container< std::vector >::container(), GiNaC::container< std::vector >::container(), GiNaC::container< std::vector >::container(), GiNaC::container< std::vector >::contai duplicate(), GiNaC::possymbol::duplicate(), GiNaC::realsymbol::duplicate(), GiNaC::power::eval(), GiNaC::add::expand(), expand(), GiNaC::expairseq::expand(), GiNaC::function::expand(), GiNaC::integral::expand(), GiNaC::mul::expand(), GiNaC::ncmul::expand(), GiNaC::power::expand(), GiNaC::power::expand_mul(), GiNaC::GINAC_IMPLEMENT_REGISTERED_CLA hold(), GiNaC::expairseq::info(), GiNaC::mul::info(), GiNaC::power::info(), GiNaC::matrix::matrix(), GiNaC::matrix::matrix(), GiNaC::matrix::matrix(), GiNaC::matrix::matrix(), GiNaC::matrix::matrix(), GiNaC::numeric::numeric(), GiNaC::numeric::numeric(), GiNaC::numeric::numeric(), GiNaC::numeric::numeric(), GiNaC::numeric::numeric(), GiNaC::numeric::numeric(), GiNaC::numeric::numeric(), GiNaC::numeric::numeric(), GiNaC::container< std::vector >::read_archive(), GiNaC::numeric::read_archive(), GiNaC::symbol::read_archive(), GiNaC::wildcard::read_archive(), GiNaC::reduced_matrix(), GiNaC::sub_matrix(), GiNaC::symbol::symbol(), GiNaC::symbol::symbol(), GiNaC::symbolic_matrix(), GiNaC::symmetry::symmetry(), GiNaC::symmetry::symmetry(), GiNaC::unit_matrix(), and GiNaC::wildcard::wildcard().

8.8.3.64 clearflag()

```
const basic & GiNaC::basic::clearflag (
            unsigned f) const  [inline]
```

Clear some status_flags.

References basic(), and flags.

Referenced by GiNaC::add::combine_ex_with_coeff_to_pair(), ensure_if_modifiable(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::function::function(), GiNaC::expairseq::info(), GiNaC::power::info(), map(), GiNaC::idx::map(), GiNaC::idx::replace_dim(), GiNaC::mul::smod(), GiNaC::add::split_ex_to_pair(), subs(), GiNaC::idx::subs(), GiNaC::spinidx::toggle_dot(), GiNaC::varidx::toggle_variance(), and GiNaC::spinidx::toggle_variance_dot().

8.8.3.65 ensure_if_modifiable()

```
void GiNaC::basic::ensure_if_modifiable () const  [protected]
```

Ensure the object may be modified without hurting others, throws if this is not the case.

References clearflag(), ensure_if_modifiable(), GiNaC::status_flags::evaluated, GiNaC::refcounted::get_refcount(), and GiNaC::status_flags::hash_calculated.

Referenced by GiNaC::container< C >::append(), GiNaC::matrix::division_free_elimination(), ensure_if_modifiable(), GiNaC::matrix::fraction_free_elimination(), GiNaC::matrix::gauss_elimination(), let_op(), GiNaC::clifford::let_op(), GiNaC::container< C >::let_op(), GiNaC::Ebar_kernel::let_op(), GiNaC::Eisenstein_h_kernel::let_op(), GiNaC::Eisenstein_kernel::let GiNaC::ELi_kernel::let_op(), GiNaC::integral::let_op(), GiNaC::Kronecker_dtau_kernel::let_op(), GiNaC::Kronecker_dz_kernel::let_op GiNaC::matrix::let_op(), GiNaC::modular_form_kernel::let_op(), GiNaC::multiple_polylog_kernel::let_op(), GiNaC::user_defined_kernel::let_op(), GiNaC::matrix::operator()(), GiNaC::matrix::pivot(), GiNaC::container< C >::prepend(), GiNaC::container< C >::remove_all(), GiNaC::container< C >::remove_first(), GiNaC::container< C >::remove_last(), GiNaC::container< C >::sort(), and GiNaC::container< C >::unique().

8.8.3.66 do_print()

```
void GiNaC::basic::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

Default output to stream.

References do_print(), and GiNaC::print_context::s.

Referenced by do_print(), and GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.8.3.67 do_print_tree()

```
void GiNaC::basic::do_print_tree (
             const print_tree & c,
             unsigned level) const  [protected]
```

Tree output to stream.

References GiNaC::print_tree::delta_indent, do_print_tree(), flags, hashvalue, nops(), op(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by do_print_tree(), GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GiNaC::GINAC_IMPLEMENT_REGIST GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), and GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.8.3.68 do_print_python_repr()

```
void GiNaC::basic::do_print_python_repr (
             const print_python_repr & c,
             unsigned level) const  [protected]
```

Python parsable output to stream.

References do_print_python_repr(), and GiNaC::print_context::s.

Referenced by do_print_python_repr(), and GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.8.4 Friends And Related Symbol Documentation

8.8.4.1 ex

```
friend class ex  [friend]
```

References ex.

Referenced by add_indexed(), GiNaC::matrix::add_indexed(), GiNaC::structure< T, ComparisonPolicy >::add_indexed(), GiNaC::pseries::add_series(), GiNaC::container< C >::append(), GiNaC::container< std::vector >::append(), GiNaC::Gt::apply_function_recursive(), GiNaC::Gt::applyIntegrationPath(), GiNaC::relational::canonical(), GiNaC::matrix::charpoly(), coeff(), GiNaC::numeric::coeff(), GiNaC::power::coeff(), GiNaC::pseries::coeff(), GiNaC::structure< T, ComparisonPolicy >::coeff(), GiNaC::pseries::coeffop(), collect(), GiNaC::pseries::collect(), GiNaC::structure< T, ComparisonPolicy >::collect(), GiNaC::expairseq::combine_ex_with_coeff_to_pair(), GiNaC::expairseq::combine_overall_coeff(), GiNaC::expairseq::combine_overal GiNaC::expairseq::combine_pair_with_coeff_to_pair(), conjugate(), GiNaC::constant::conjugate(), GiNaC::container< std::vector >::c GiNaC::expairseq::conjugate(), GiNaC::integral::conjugate(), GiNaC::matrix::conjugate(), GiNaC::numeric::conjugate(), GiNaC::power::conjugate(), GiNaC::pseries::conjugate(), GiNaC::symbol::conjugate(), GiNaC::expairseq::construct_from_2_ex(), GiNaC::expairseq::construct_from_expairseq_ex(), GiNaC::pseries::convert_to_poly(), GiNaC::Gt::decomposeIntegrationPath(), GiNaC::expairseq::default_overall_coeff(), degree(), GiNaC::integral::degree(), GiNaC::numeric::degree(), GiNaC::power::degree(), GiNaC::pseries::degree(), GiNaC::structure< T, ComparisonPolicy >::degree(), derivative(), GiNaC::constant::derivative(), GiNaC::idx::derivative(), GiNaC::integral::derivative(), GiNaC::numeric::derivative(), GiNaC::power::derivative(), GiNaC::pseries::derivative(), GiNaC::structure< T, ComparisonPolicy >::derivative(), GiNaC::symbol::derivative(), GiNaC::matrix::determinant(), GiNaC::matrix::determinant_minor(), diff(), eval(), GiNaC::expairseq::eval(), GiNaC::Gt::eval(), GiNaC::integral::eval(), GiNaC::numeric::eval(), GiNaC::power::eval(), GiNaC::pseries::eval(), GiNaC::structure< T, ComparisonPolicy >::eval(), GiNaC::symbol::eval(), eval_indexed(), GiNaC::matrix::eval_indexed(), GiNaC::structure< T, ComparisonPolicy >::eval_indexed(), eval_integ(), GiNaC::integral::eval_integ(), GiNaC::pseries::eval_integ(), eval_ncmul(), GiNaC::integral::eval_ncmul(), GiNaC::power::eval_ncmul(), GiNaC::relational::eval_ncmu GiNaC::structure< T, ComparisonPolicy >::eval_ncmul(), evalf(), GiNaC::constant::evalf(), GiNaC::Gt::evalf(), GiNaC::idx::evalf(), GiNaC::integral::evalf(), GiNaC::numeric::evalf(), GiNaC::power::evalf(), GiNaC::pseries::evalf(), GiNaC::symbol::evalf(), evalm(), GiNaC::matrix::evalm(), GiNaC::power::evalm(), GiNaC::pseries::evalm(), GiNaC::structure< T, ComparisonPolicy >::evalm(), GiNaC::Gt::evaluate(), ex, GiNaC::Gt::ex_cutIntegrationPath(), GiNaC::Gt::ex_cutIntegrationPath(), GiNaC::Gt::ex_evaluate(), GiNaC::Gt::ex_evaluate(), GiNaC::Gt::ex_prepare(), GiNaC::Gt::ex_prepare(), GiNaC::Gt::ex_qExpand(), GiNaC::Gt::ex_qExpand(), GiNaC::Gt::ex_regularise(), GiNaC::Gt::ex_regularise(), GiNaC::Gt::ex_tauToFundamental(), GiNaC::Gt::ex_tauToFundamental(), GiNaC::Gt::ex_zisToFundamen GiNaC::Gt::ex_zisToFundamental(), GiNaC::expairseq::expairseq(), GiNaC::expairseq::expairseq(), GiNaC::expairseq::expairseq(), expand(), GiNaC::expairseq::expand(), GiNaC::integral::expand(), GiNaC::power::expand(), GiNaC::pseries::expand(), GiNaC::structure< T, ComparisonPolicy >::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), GiNaC::power::expand_mul(), GiNaC::expairseq::expandchildren(), GiNaC::pseries::exponop(), GiNaC::Gt::findMoebiusTransform(), GiNaC::matrix::fraction_free_elimination(), GiNaC::matrix::gauss_elimination(), GiNaC::idx::get_dim(), GiNaC::integration_kernel::get GiNaC::integration_kernel::get_numerical_value_impl(), GiNaC::pseries::get_point(), GiNaC::integration_kernel::get_series_coeff(), GiNaC::idx::get_value(), GiNaC::pseries::get_var(), GiNaC::Gt::getTau(), GiNaC::Gt::getZ(), GiNaC::Gt::Gt(), GiNaC::Gt::Gt(), has(), GiNaC::Gt::has(), GiNaC::numeric::has(), GiNaC::power::has(), GiNaC::structure< T, ComparisonPolicy >::ha GiNaC::idx::idx(), imag_part(), GiNaC::constant::imag_part(), GiNaC::container< std::vector >::imag_part(), GiNaC::matrix::imag_part(), GiNaC::numeric::imag_part(), GiNaC::power::imag_part(), GiNaC::pseries::imag_part(), GiNaC::symbol::imag_part(), GiNaC::integral::integral(), is_polynomial(), GiNaC::constant::is_polynomial(), GiNaC::numeric::is_polynomial(), GiNaC::power::is_polynomial(), GiNaC::symbol::is_polynomial(), GiNaC::Gt::kernel::kernel(), GiNaC::Gt::kernel::kernel(), GiNaC::integration_kernel::Laurent_series(), ldegree(), GiNaC::integral::ldegree(), GiNaC::numeric::ldegree(), GiNaC::power::ldegree(), GiNaC::pseries::ldegree(), GiNaC::structure< T, ComparisonPolicy >::ldegree(let_op(), GiNaC::container< C >::let_op(), GiNaC::container< std::vector >::let_op(), GiNaC::integral::let_op(), GiNaC::matrix::let_op(), GiNaC::structure< T, ComparisonPolicy >::let_op(), GiNaC::relational::lhs(), GiNaC::Gt::lst_evaluate(), GiNaC::Gt::lst_evaluate(), GiNaC::expairseq::make_flat(), GiNaC::expairseq::make_flat(), map(), GiNaC::expairseq::map(), GiNaC::idx::map(), GiNaC::power::map(), GiNaC::relational::map(), GiNaC::structure< T, ComparisonPolicy >::map(), GiNaC::matrix::markowitz_elimination(), match(), GiNaC::expairseq::match(), GiNaC::Gt::match(), GiNaC::structure< T, ComparisonP GiNaC::wildcard::match(), GiNaC::idx::minimal_dim(), GiNaC::pseries::mul_const(), GiNaC::matrix::mul_scalar(), GiNaC::pseries::mul_series(), normal(), GiNaC::numeric::normal(), GiNaC::power::normal(), GiNaC::pseries::normal(), GiNaC::structure< T, ComparisonPolicy >::normal(), GiNaC::symbol::normal(), op(), GiNaC::container< C >::op(), GiNaC::container< std::vector >::op(), GiNaC::expairseq::op(), GiNaC::idx::op(), GiNaC::integral::op(), GiNaC::matrix::op(),

GiNaC::power::op(), GiNaC::pseries::op(), GiNaC::relational::op(), GiNaC::structure< T, ComparisonPolicy >::op(), GiNaC::relational::operator safe_bool(), GiNaC::matrix::operator()(), GiNaC::matrix::operator()(), operator[](), operator[](), operator[](), operator[](), GiNaC::structure< T, ComparisonPolicy >::operator[](), GiNaC::structure< T, ComparisonPoli GiNaC::structure< T, ComparisonPolicy >::operator[](), GiNaC::structure< T, ComparisonPolicy >::operator[](), GiNaC::matrix::pow(), GiNaC::power::power(), GiNaC::power::power(), GiNaC::pseries::power_const(), GiNaC::container< C >::prep GiNaC::container< std::vector >::prepend(), GiNaC::pseries::pseries(), GiNaC::pseries::pseries(), GiNaC::Gt::qExpand(), GiNaC::container< std::vector >::read_archive(), GiNaC::expairseq::read_archive(), GiNaC::Gt::read_archive(), GiNaC::matrix::read_archive(), GiNaC::pseries::read_archive(), GiNaC::symmetry::read_archive(), real_part(), GiNaC::constant::real_part(), GiNaC::container< std::vector >::real_part(), GiNaC::matrix::real_part(), GiNaC::numeric::real_part(), GiNaC::power::real_part(), GiNaC::pseries::real_part(), GiNaC::symbol::real_part(), GiNaC::expairseq::recombine_pair_to_ex(), GiNaC::Gt::regularise(), GiNaC::relational::relational(), GiNaC::tensor::replace_contr_index(), GiNaC::idx::replace_dim(), GiNaC::relational::rhs(), scalar_mul_indexed(), GiNaC::matrix::scalar_mul_indexed(), GiNaC::structure< T, ComparisonPolicy >::sca series(), GiNaC::integral::series(), GiNaC::integration_kernel::series(), GiNaC::power::series(), GiNaC::pseries::series(), GiNaC::structure< T, ComparisonPolicy >::series(), GiNaC::symbol::series(), GiNaC::integration_kernel::series_coeff(), GiNaC::matrix::set(), GiNaC::Gt::setArgs(), smod(), GiNaC::numeric::smod(), GiNaC::structure< T, ComparisonPolicy >::smod(), GiNaC::matrix::solve(), GiNaC::expairseq::split_ex_to_pair(), subs(), GiNaC::container< C >::subs(), GiNaC::container< std::vector GiNaC::expairseq::subs(), GiNaC::Gt::subs(), GiNaC::idx::subs(), GiNaC::matrix::subs(), GiNaC::numeric::subs(), GiNaC::power::subs(), GiNaC::pseries::subs(), GiNaC::relational::subs(), GiNaC::structure< T, ComparisonPolicy >::subs(), GiNaC::symbol::subs(), subs_one_level(), GiNaC::container< C >::subschildren(), GiNaC::expairseq::subschildren(), GiNaC::Gt::tauToFundamental(), GiNaC::container< std::vector >::thiscontainer(), GiNaC::container< std::vector >::thiscontainer(), GiNaC::expairseq::thisexpairseq(), GiNaC::expairseq::thisexpairseq(), GiNaC::Gt::to_integer(), GiNaC::Gt::to_numeric(), to_polynomial(), GiNaC::expairseq::to_polynomial(), GiNaC::numeric::to_polynomial(), GiNaC::power::to_polynomial(), GiNaC::structure< T, ComparisonPolicy >::to_polynomial(), GiNaC::symbol::to_polynomial(), to_rational(), GiNaC::expairseq::to_rational(), GiNaC::numeric::to_rational(), GiNaC::power::to_rational(), GiNaC::structure< T, ComparisonPolicy GiNaC::symbol::to_rational(), GiNaC::matrix::trace(), and GiNaC::Gt::zisToFundamental().

8.8.5 Member Data Documentation

8.8.5.1 flags

```
unsigned GiNaC::basic::flags  [mutable], [protected]
```

of type status_flags

Referenced by basic(), calchash(), GiNaC::expairseq::calchash(), GiNaC::function::calchash(), GiNaC::Gt::calchash(), GiNaC::idx::calchash(), GiNaC::relational::calchash(), GiNaC::symmetry::calchash(), clearflag(), GiNaC::ex::construct_from_basic(), diff(), do_print_tree(), GiNaC::clifford::do_print_tree(), GiNaC::constant::do_print_tree(), GiNaC::container< C >::do_print_tree(), GiNaC::expairseq::do_print_tree(), GiNaC::fderivative::do_print_tree(), GiNaC::idx::do_print_tree(), GiNaC::indexed::do_print_tree(), GiNaC::numeric::do_print_tree(), GiNaC::pseries::do_print_tree(), GiNaC::spinidx::do_print_tree(), GiNaC::symbol::do_print_tree(), GiNaC::symmetry::do_print_tree(), GiNaC::varidx::do_print_tree(), GiNaC::wildcard::do_print_tree(), GiNaC::add::eval(), GiNaC::expairseq::eval(), GiNaC::function::eval(), GiNaC::Gt::eval(), GiNaC::integral::eval(), GiNaC::mul::eval(), GiNaC::ncmul::eval(), GiNaC::power::eval(), GiNaC::integral::eval_integ(), GiNaC::integral::expand(), gethash(), GiNaC::expairseq::info(), GiNaC::mul::info(), GiNaC::power::info(), operator=(), GiNaC::function::print(), setflag(), and ~basic().

8.8.5.2 hashvalue

```
unsigned GiNaC::basic::hashvalue  [mutable], [protected]
```

hash value

Referenced by calchash(), GiNaC::constant::calchash(), GiNaC::expairseq::calchash(), GiNaC::function::calchash(), GiNaC::Gt::calchash(), GiNaC::idx::calchash(), GiNaC::numeric::calchash(), GiNaC::relational::calchash(), GiNaC::symbol::calchash(), GiNaC::symmetry::calchash(), GiNaC::wildcard::calchash(), do_print_tree(), GiNaC::clifford::do_print_tree

GiNaC::constant::do_print_tree(), GiNaC::container< C >::do_print_tree(), GiNaC::expairseq::do_print_tree(), GiNaC::fderivative::do_print_tree(), GiNaC::idx::do_print_tree(), GiNaC::indexed::do_print_tree(), GiNaC::numeric::do_print_tree(), GiNaC::pseries::do_print_tree(), GiNaC::spinidx::do_print_tree(), GiNaC::symbol::do_print_tree(), GiNaC::symmetry::do_print_tree(), GiNaC::varidx::do_print_tree(), GiNaC::wildcard::do_print_tree(), gethash(), operator=(), and GiNaC::function::print().

The documentation for this class was generated from the following files:

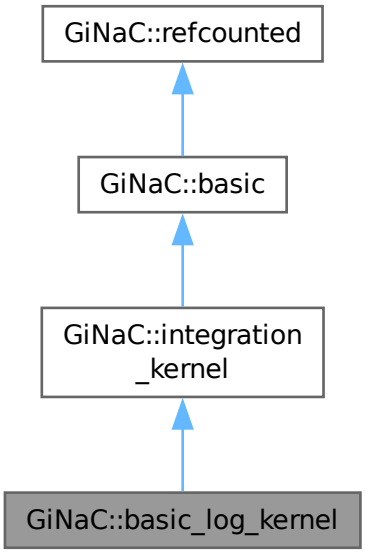
- basic.h
- basic.cpp
- normal.cpp
- pseries.cpp

8.9 GiNaC::basic_log_kernel Class Reference

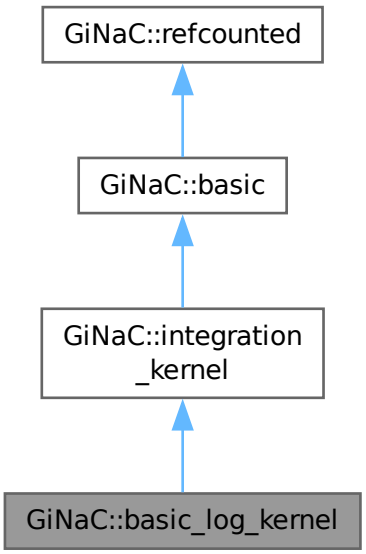
The basic integration kernel with a logarithmic singularity at the origin.

`#include <integration_kernel.h>`

Inheritance diagram for GiNaC::basic_log_kernel:



Collaboration diagram for GiNaC::basic_log_kernel:



Protected Member Functions

- cln::cl_N series_coeff_impl (int i) const override

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual bool uses_Laurent_series () const

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.

- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Public Member Functions inherited from GiNaC::integration_kernel

- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- virtual bool is_numeric (void) const

  This routine returns true, if the integration kernel can be evaluated numerically.
- virtual ex Laurent_series (const ex &x, int order) const

  Returns the Laurent series, starting possibly with the pole term.
- virtual ex get_numerical_value (const ex &lambda, int N_trunc=0) const

  Evaluates the integrand at lambda.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.

- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().

- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags
  of type status_flags
- unsigned hashvalue
  hash value

8.9.1 Detailed Description

The basic integration kernel with a logarithmic singularity at the origin.

This class represents the differential one-form

$$L_0 = \frac{d\lambda}{\lambda}$$

8.9.2 Member Function Documentation

8.9.2.1 series_coeff_impl()

```
cln::cl_N GiNaC::basic_log_kernel::series_coeff_impl (
            int i) const  [override], [protected], [virtual]
```

For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.

The i-th coefficient corresponds to the power λ^{i-1}.

Reimplemented from GiNaC::integration_kernel.

References series_coeff_impl().

Referenced by series_coeff_impl().

8.9.2.2 do_print()

```
void GiNaC::basic_log_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References do_print(), and GiNaC::print_context::s.

Referenced by do_print(), and GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

The documentation for this class was generated from the following files:

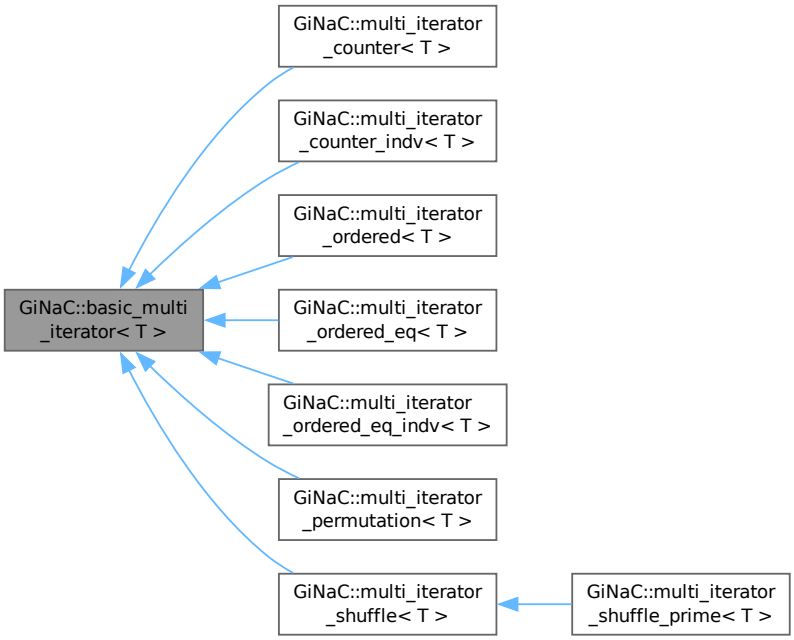
- integration_kernel.h
- integration_kernel.cpp

8.10 GiNaC::basic_multi_iterator< T > Class Template Reference

basic_multi_iterator is a base class.

`#include <utils_multi_iterator.h>`

Inheritance diagram for GiNaC::basic_multi_iterator< T >:



Public Member Functions

- basic_multi_iterator (void)
  Default constructor.
- basic_multi_iterator (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N, lower limit B and size k .
- basic_multi_iterator (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- virtual ~basic_multi_iterator ()
  Destructor.
- size_t size (void) const
  Returns the size of a multi_iterator.
- bool overflow (void) const
  Return the overflow flag.
- const std::vector< T > & get_vector (void) const
  Returns a reference to the vector v.
- T operator[ ] (size_t i) const
  Subscription via [].
- T & operator[ ] (size_t i)

Subscription via [].

- T operator() (size_t i) const

  Subscription via ().

- T & operator() (size_t i)

  Subscription via ().

- virtual basic_multi_iterator< T > & init (void)

  Initialize the multi-index to.

- virtual basic_multi_iterator< T > & operator++ (int)

  No effect for basic_multi_iterator.

Protected Attributes

- T N
- T B
- std::vector< T > v
- bool flag_overflow

Friends

- template<class TT>
  std::ostream & operator<< (std::ostream &os, const basic_multi_iterator< TT > &v)

8.10.1 Detailed Description

template<class T>
class GiNaC::basic_multi_iterator< T >

basic_multi_iterator is a base class.

The base class itself does not do anything useful. A typical use of a class derived from basic_multi_iterator is

multi_iterator_ordered<int> k(0,4,2);

for(k.init(); !k.overflow(); k++) { std::cout << k << std::endl; }

which prints out

multi_iterator_ordered(0,1) multi_iterator_ordered(0,2) multi_iterator_ordered(0,3) multi_iterator_ordered(1,2) multi_iterator_ordered(1,3) multi_iterator_ordered(2,3)

Individual components of k can be accessed with k[i] or k(i).

All classes derived from basic_multi_iterator follow the same syntax.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 basic_multi_iterator() [1/3]

```
template<class T>
GiNaC::basic_multi_iterator< T >::basic_multi_iterator (
            void )  [inline]
```

Default constructor.

References B, flag_overflow, N, and v.

Referenced by init(), GiNaC::multi_iterator_counter< T >::init(), GiNaC::multi_iterator_counter_indv< T >::init(), GiNaC::multi_iterator_ordered< T >::init(), GiNaC::multi_iterator_ordered_eq< T >::init(), GiNaC::multi_iterator_ordered_eq_indv< GiNaC::multi_iterator_permutation< T >::init(), GiNaC::multi_iterator_shuffle< T >::init(), GiNaC::multi_iterator_counter< T >::multi GiNaC::multi_iterator_counter< T >::multi_iterator_counter(), GiNaC::multi_iterator_counter< T >::multi_iterator_counter(), GiNaC::multi_iterator_counter_indv< T >::multi_iterator_counter_indv(), GiNaC::multi_iterator_counter_indv< T >::multi_iterator_co GiNaC::multi_iterator_counter_indv< T >::multi_iterator_counter_indv(), GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered(), GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered(), GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq(), GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordere GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq(), GiNaC::multi_iterator_ordered_eq_indv< T >::multi_iterator_c GiNaC::multi_iterator_ordered_eq_indv< T >::multi_iterator_ordered_eq_indv(), GiNaC::multi_iterator_ordered_eq_indv< T >::mult GiNaC::multi_iterator_permutation< T >::multi_iterator_permutation(), GiNaC::multi_iterator_permutation< T >::multi_iterator_perm GiNaC::multi_iterator_permutation< T >::multi_iterator_permutation(), GiNaC::multi_iterator_shuffle< T >::multi_iterator_shuffle(), GiNaC::multi_iterator_shuffle< T >::multi_iterator_shuffle(), operator++(), GiNaC::multi_iterator_counter< T >::operator++(), GiNaC::multi_iterator_counter_indv< T >::operator++(), GiNaC::multi_iterator_ordered< T >::operator++(), GiNaC::multi_iterator_ordered_eq< T >::operator++(), GiNaC::multi_iterator_ordered_eq_indv< T >::operator++(), GiNaC::multi_iterator_permutation< T >::operator++(), GiNaC::multi_iterator_shuffle< T >::operator++(), and operator<<.

8.10.2.2 basic_multi_iterator() [2/3]

```
template<class T>
GiNaC::basic_multi_iterator< T >::basic_multi_iterator (
            T B,
            T N,
            size_t k)  [inline], [explicit]
```

Construct a multi_iterator with upper limit N, lower limit B and size k .

References B, flag_overflow, N, and v.

8.10.2.3 basic_multi_iterator() [3/3]

```
template<class T>
GiNaC::basic_multi_iterator< T >::basic_multi_iterator (
            T B,
            T N,
            const std::vector< T > & vv)  [inline], [explicit]
```

Construct from a vector.

References B, flag_overflow, N, and v.

8.10.2.4 ~basic_multi_iterator()

```
template<class T>
GiNaC::basic_multi_iterator< T >::~basic_multi_iterator ()  [inline], [virtual]
```

Destructor.

8.10.3 Member Function Documentation

8.10.3.1 size()

```
template<class T>
size_t GiNaC::basic_multi_iterator< T >::size (
            void ) const  [inline]
```

Returns the size of a multi_iterator.

References v.

Referenced by GiNaC::multi_iterator_permutation< T >::get_sign(), GiNaC::multi_iterator_counter< T >::operator++(), GiNaC::multi_iterator_counter_indv< T >::operator++(), GiNaC::multi_iterator_ordered< T >::operator++(), GiNaC::multi_iterator_ordered_eq< T >::operator++(), GiNaC::multi_iterator_ordered_eq_indv< T >::operator++(), GiNaC::multi_iterator_permutation< T >::operator++(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(), and GiNaC::operator<<().

8.10.3.2 overflow()

```
template<class T>
bool GiNaC::basic_multi_iterator< T >::overflow (
            void ) const  [inline]
```

Return the overflow flag.

References flag_overflow.

Referenced by GiNaC::Gt::regularise().

8.10.3.3 get_vector()

```
template<class T>
const std::vector< T > & GiNaC::basic_multi_iterator< T >::get_vector (
            void ) const  [inline]
```

Returns a reference to the vector v.

References v.

Referenced by GiNaC::Gt::regularise().

8.10.3.4 operator[]() [1/2]

```
template<class T>
T GiNaC::basic_multi_iterator< T >::operator[] (
            size_t i) const  [inline]
```

Subscription via [].

References v.

8.10.3.5 operator[]() [2/2]

```
template<class T>
T & GiNaC::basic_multi_iterator< T >::operator[] (
            size_t i)  [inline]
```

Subscription via [].

References v.

8.10.3.6 operator()() [1/2]

```
template<class T>
T GiNaC::basic_multi_iterator< T >::operator() (
            size_t i) const  [inline]
```

Subscription via ().

References v.

8.10.3.7 operator()() [2/2]

```
template<class T>
T & GiNaC::basic_multi_iterator< T >::operator() (
            size_t i)  [inline]
```

Subscription via ().

References v.

8.10.3.8 init()

```
template<class T>
basic_multi_iterator< T > & GiNaC::basic_multi_iterator< T >::init (
            void )  [inline], [virtual]
```

Initialize the multi-index to.

$$(n_1, n_2, n_3, ..., n_k) = (B, B, ..., B)$$

Reimplemented in GiNaC::multi_iterator_counter< T >, GiNaC::multi_iterator_counter_indv< T >, GiNaC::multi_iterator_ordered< T
GiNaC::multi_iterator_ordered_eq< T >, GiNaC::multi_iterator_ordered_eq_indv< T >, GiNaC::multi_iterator_permutation< T >,
GiNaC::multi_iterator_shuffle< T >, and GiNaC::multi_iterator_shuffle_prime< T >.

References B, basic_multi_iterator(), flag_overflow, and v.

8.10.3.9 operator++()

```
template<class T>
basic_multi_iterator< T > & GiNaC::basic_multi_iterator< T >::operator++ (
            int )  [inline], [virtual]
```

No effect for basic_multi_iterator.

Reimplemented in GiNaC::multi_iterator_counter< T >, GiNaC::multi_iterator_counter_indv< T >, GiNaC::multi_iterator_ordered< T
GiNaC::multi_iterator_ordered_eq< T >, GiNaC::multi_iterator_ordered_eq_indv< T >, GiNaC::multi_iterator_permutation< T >,
and GiNaC::multi_iterator_shuffle< T >.

References basic_multi_iterator().

8.10.4 Friends And Related Symbol Documentation

8.10.4.1 operator<<

```
template<class T>
template<class TT>
std::ostream & operator<< (
            std::ostream & os,
            const basic_multi_iterator< TT > & v)  [friend]
```

References basic_multi_iterator(), and v.

8.10.5 Member Data Documentation

8.10.5.1 N

```
template<class T>
T GiNaC::basic_multi_iterator< T >::N  [protected]
```

Referenced by basic_multi_iterator(), basic_multi_iterator(), basic_multi_iterator(), GiNaC::multi_iterator_counter< T >::multi_iterator
GiNaC::multi_iterator_counter< T >::multi_iterator_counter(), GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered(),
GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered(), GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq(),
GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq(), GiNaC::multi_iterator_permutation< T >::multi_iterator_permu
GiNaC::multi_iterator_permutation< T >::multi_iterator_permutation(), GiNaC::multi_iterator_counter< T >::operator++(),
GiNaC::multi_iterator_ordered< T >::operator++(), GiNaC::multi_iterator_ordered_eq< T >::operator++(), and
GiNaC::multi_iterator_permutation< T >::operator++().

8.10.5.2 B

```
template<class T>
T GiNaC::basic_multi_iterator< T >::B  [protected]
```

Referenced by basic_multi_iterator(), basic_multi_iterator(), basic_multi_iterator(), init(), GiNaC::multi_iterator_counter< T >::init(),
GiNaC::multi_iterator_counter_indv< T >::init(), GiNaC::multi_iterator_ordered< T >::init(), GiNaC::multi_iterator_ordered_eq< T >
GiNaC::multi_iterator_ordered_eq_indv< T >::init(), GiNaC::multi_iterator_permutation< T >::init(), GiNaC::multi_iterator_counter<
GiNaC::multi_iterator_counter< T >::multi_iterator_counter(), GiNaC::multi_iterator_counter_indv< T >::multi_iterator_counter_indv
GiNaC::multi_iterator_counter_indv< T >::multi_iterator_counter_indv(), GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered
GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered(), GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq(),
GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq(), GiNaC::multi_iterator_ordered_eq_indv< T >::multi_iterator_c
GiNaC::multi_iterator_ordered_eq_indv< T >::multi_iterator_ordered_eq_indv(), GiNaC::multi_iterator_permutation< T >::multi_iter
GiNaC::multi_iterator_permutation< T >::multi_iterator_permutation(), GiNaC::multi_iterator_shuffle< T >::multi_iterator_shuffle(),
GiNaC::multi_iterator_counter< T >::operator++(), GiNaC::multi_iterator_counter_indv< T >::operator++(),
GiNaC::multi_iterator_permutation< T >::operator++(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(),
GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(),
and GiNaC::operator<<().

8.10.5.3 v

```
template<class T>
std::vector<T> GiNaC::basic_multi_iterator< T >::v  [protected]
```

Referenced by basic_multi_iterator(), basic_multi_iterator(), basic_multi_iterator(), GiNaC::multi_iterator_permutation< T >::get_sign get_vector(), init(), GiNaC::multi_iterator_counter< T >::init(), GiNaC::multi_iterator_counter_indv< T >::init(), GiNaC::multi_iterator_ordered< T >::init(), GiNaC::multi_iterator_ordered_eq< T >::init(), GiNaC::multi_iterator_ordered_eq_indv< GiNaC::multi_iterator_permutation< T >::init(), GiNaC::multi_iterator_shuffle< T >::init(), GiNaC::multi_iterator_shuffle_prime< T > GiNaC::multi_iterator_counter< T >::multi_iterator_counter(), GiNaC::multi_iterator_counter_indv< T >::multi_iterator_counter_indv GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered(), GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq(), GiNaC::multi_iterator_ordered_eq_indv< T >::multi_iterator_ordered_eq_indv(), GiNaC::multi_iterator_permutation< T >::multi_iter GiNaC::multi_iterator_shuffle< T >::multi_iterator_shuffle(), operator()(), operator()(), GiNaC::multi_iterator_counter< T >::operator+ GiNaC::multi_iterator_counter_indv< T >::operator++(), GiNaC::multi_iterator_ordered< T >::operator++(), GiNaC::multi_iterator_ordered_eq< T >::operator++(), GiNaC::multi_iterator_ordered_eq_indv< T >::operator++(), GiNaC::multi_iterator_permutation< T >::operator++(), GiNaC::multi_iterator_shuffle< T >::operator++(), operator<<, GiNaC::multi_iterator_counter< T >::operator<<, GiNaC::multi_iterator_counter_indv< T >::operator<<, GiNaC::multi_iterator_ordered< T >::operator<<, GiNaC::multi_iterator_ordered_eq< T >::operator<<, GiNaC::multi_iterator_ordered_eq_indv< T >::operator<<, GiNaC::multi_iterator_permutation< T >::operator<<, GiNaC::multi_iterator_shuffle< T >::operator<<, GiNaC::multi_iterator_shuffle_prime< T >::operator<<, operator[ ](), operator[ ](), and size().

8.10.5.4 flag_overflow

```
template<class T>
bool GiNaC::basic_multi_iterator< T >::flag_overflow  [protected]
```

Referenced by basic_multi_iterator(), basic_multi_iterator(), basic_multi_iterator(), init(), GiNaC::multi_iterator_counter< T >::init(), GiNaC::multi_iterator_counter_indv< T >::init(), GiNaC::multi_iterator_ordered< T >::init(), GiNaC::multi_iterator_ordered_eq< T > GiNaC::multi_iterator_ordered_eq_indv< T >::init(), GiNaC::multi_iterator_permutation< T >::init(), GiNaC::multi_iterator_shuffle< T GiNaC::multi_iterator_shuffle_prime< T >::init(), GiNaC::multi_iterator_counter< T >::operator++(), GiNaC::multi_iterator_counter_i GiNaC::multi_iterator_ordered< T >::operator++(), GiNaC::multi_iterator_ordered_eq< T >::operator++(), GiNaC::multi_iterator_ordered_eq_indv< T >::operator++(), GiNaC::multi_iterator_permutation< T >::operator++(), GiNaC::multi_iterator_shuffle< T >::operator++(), and overflow().

The documentation for this class was generated from the following file:

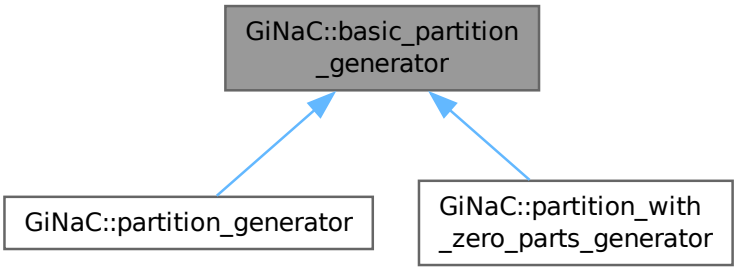
- utils_multi_iterator.h

8.11 GiNaC::basic_partition_generator Class Reference

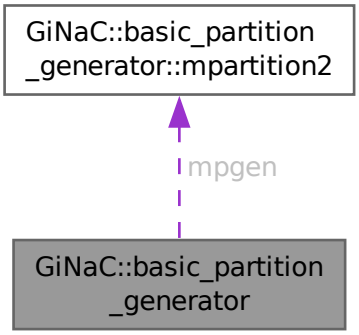
Base class for generating all bounded combinatorial partitions of an integer n with exactly m parts in non-decreasing order.

```
#include <utils.h>
```

Inheritance diagram for GiNaC::basic_partition_generator:



Collaboration diagram for GiNaC::basic_partition_generator:



Classes

- struct mpartition2

Protected Member Functions

- basic_partition_generator (unsigned n_, unsigned m_)

Protected Attributes

- mpartition2 mpgen

8.11.1 Detailed Description

Base class for generating all bounded combinatorial partitions of an integer n with exactly m parts in non-decreasing order.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 basic_partition_generator()

```
GiNaC::basic_partition_generator::basic_partition_generator (
            unsigned n_,
            unsigned m_)  [inline], [protected]
```

References mpgen.

Referenced by GiNaC::partition_generator::partition_generator(), and GiNaC::partition_with_zero_parts_generator::partition_with_zer

8.11.3 Member Data Documentation

8.11.3.1 mpgen

```
mpartition2 GiNaC::basic_partition_generator::mpgen  [protected]
```

Referenced by basic_partition_generator(), GiNaC::partition_generator::get(), GiNaC::partition_with_zero_parts_generator::get(), GiNaC::partition_generator::next(), and GiNaC::partition_with_zero_parts_generator::next().

The documentation for this class was generated from the following file:

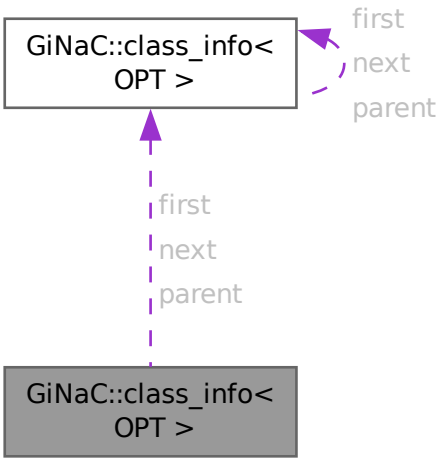
- utils.h

8.12 GiNaC::class_info< OPT > Class Template Reference

```
#include <class_info.h>
```

Inheritance diagram for GiNaC::class_info< OPT >:

Collaboration diagram for GiNaC::class_info< OPT >:



Classes

- struct tree_node

Public Member Functions

- class_info (const OPT &o)
- class_info * get_parent () const

  Get pointer to class_info of parent class (or nullptr).

Static Public Member Functions

- static const class_info * find (const std::string &class_name)

  Find class_info by name.
- static void dump_hierarchy (bool verbose=false)

  Dump class hierarchy to std::cout.

Public Attributes

- OPT options

Static Private Member Functions

- static void dump_tree (tree_node *n, const std::string &prefix, bool verbose)
- static void identify_parents ()

Private Attributes

- class_info * next
- class_info * parent

Static Private Attributes

- static class_info * first
- static bool parents_identified

8.12.1 Constructor & Destructor Documentation

8.12.1.1 class_info()

```
template<class OPT>
GiNaC::class_info< OPT >::class_info (
            const OPT & o)  [inline]
```

Referenced by dump_hierarchy(), find(), identify_parents(), and GiNaC::class_info< OPT >::tree_node::tree_node().

8.12.2 Member Function Documentation

8.12.2.1 get_parent()

```
template<class OPT>
class_info * GiNaC::class_info< OPT >::get_parent () const  [inline]
```

Get pointer to class_info of parent class (or nullptr).

Referenced by GiNaC::function::print(), and GiNaC::basic::print_dispatch().

8.12.2.2 find()

```
template<class OPT>
const class_info< OPT > * GiNaC::class_info< OPT >::find (
            const std::string & class_name)  [static]
```

Find class_info by name.

References class_info(), find(), first, next, and options.

Referenced by find().

8.12.2.3 dump_hierarchy()

```
template<class OPT>
void GiNaC::class_info< OPT >::dump_hierarchy (
            bool verbose = false)  [static]
```

Dump class hierarchy to std::cout.

References class_info(), dump_tree(), first, identify_parents(), and GiNaC::tree().

8.12.2.4 dump_tree()

```
template<class OPT>
void GiNaC::class_info< OPT >::dump_tree (
            tree_node * n,
            const std::string & prefix,
            bool verbose)  [static], [private]
```

References GiNaC::class_info< OPT >::tree_node::children, dump_tree(), GiNaC::class_info< OPT >::tree_node::info, and options.

Referenced by dump_hierarchy(), and dump_tree().

8.12.2.5 identify_parents()

```
template<class OPT>
void GiNaC::class_info< OPT >::identify_parents ()  [static], [private]
```

References class_info(), first, and parents_identified.

Referenced by dump_hierarchy().

8.12.3 Member Data Documentation

8.12.3.1 options

```
template<class OPT>
OPT GiNaC::class_info< OPT >::options
```

Referenced by dump_tree(), find(), GiNaC::function::print(), and GiNaC::basic::print_dispatch().

8.12.3.2 first

```
template<class OPT>
class_info* GiNaC::class_info< OPT >::first  [static], [private]
```

Referenced by dump_hierarchy(), find(), and identify_parents().

8.12.3.3 next

```
template<class OPT>
class_info* GiNaC::class_info< OPT >::next  [private]
```

Referenced by find().

8.12.3.4 parent

```
template<class OPT>
class_info* GiNaC::class_info< OPT >::parent  [mutable], [private]
```

8.12.3.5 parents_identified

```
template<class OPT>
bool GiNaC::class_info< OPT >::parents_identified  [static], [private]
```

Referenced by identify_parents().

The documentation for this class was generated from the following file:

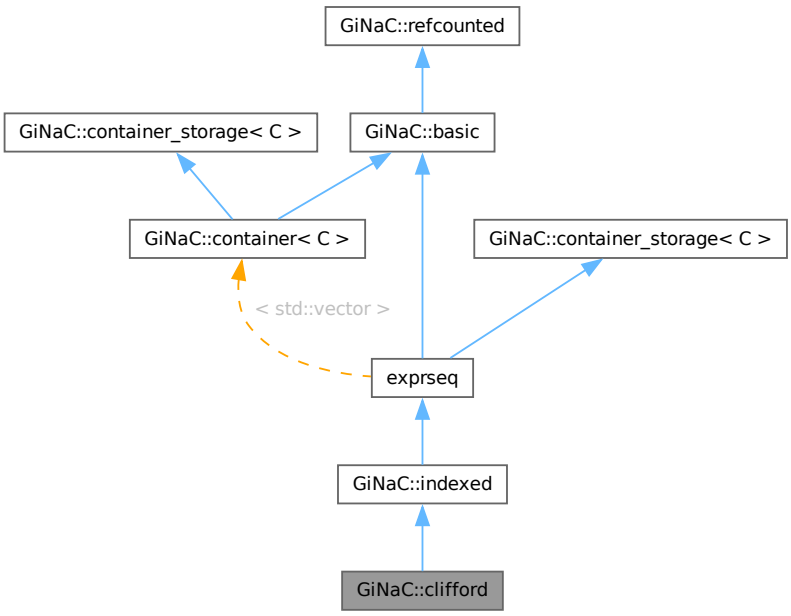
- class_info.h

8.13 GiNaC::clifford Class Reference

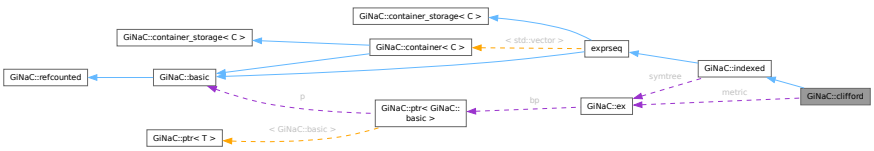
This class holds an object representing an element of the Clifford algebra (the Dirac gamma matrices).

```
#include <clifford.h>
```

Inheritance diagram for GiNaC::clifford:



Collaboration diagram for GiNaC::clifford:



Public Member Functions

- clifford (const ex &b, unsigned char rl=0)

  Construct object without any indices.
- clifford (const ex &b, const ex &mu, const ex &metr, unsigned char rl=0, int comm_sign=-1)

  Construct object with one Lorentz index.
- clifford (unsigned char rl, const ex &metr, int comm_sign, const exvector &v)
- clifford (unsigned char rl, const ex &metr, int comm_sign, exvector &&v)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- void archive (archive_node &n) const override

  Save (serialize) the object into archive node.
- void read_archive (const archive_node &n, lst &sym_lst) override

  Load (deserialize) the object from an archive node.
- unsigned char get_representation_label () const
- ex get_metric () const
- virtual ex get_metric (const ex &i, const ex &j, bool symmetrised=false) const
- bool same_metric (const ex &other) const
- int get_commutator_sign () const
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.

Public Member Functions inherited from GiNaC::indexed

- indexed (const ex &b)

  Construct indexed object with no index.
- indexed (const ex &b, const ex &i1)

  Construct indexed object with one index.
- indexed (const ex &b, const ex &i1, const ex &i2)

  Construct indexed object with two indices.
- indexed (const ex &b, const ex &i1, const ex &i2, const ex &i3)

  Construct indexed object with three indices.
- indexed (const ex &b, const ex &i1, const ex &i2, const ex &i3, const ex &i4)

  Construct indexed object with four indices.
- indexed (const ex &b, const symmetry &symm, const ex &i1, const ex &i2)

  Construct indexed object with two indices and a specified symmetry.
- indexed (const ex &b, const symmetry &symm, const ex &i1, const ex &i2, const ex &i3)

  Construct indexed object with three indices and a specified symmetry.
- indexed (const ex &b, const symmetry &symm, const ex &i1, const ex &i2, const ex &i3, const ex &i4)

  Construct indexed object with four indices and a specified symmetry.
- indexed (const ex &b, const exvector &iv)

  Construct indexed object with a specified vector of indices.
- indexed (const ex &b, const symmetry &symm, const exvector &iv)

  Construct indexed object with a specified vector of indices and symmetry.
- indexed (const symmetry &symm, const exprseq &es)

- indexed (const symmetry &symm, const exvector &v)
- indexed (const symmetry &symm, exvector &&v)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex real_part () const override
- ex imag_part () const override
- exvector get_free_indices () const override

  Return a vector containing the free indices of an expression.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- bool all_index_values_are (unsigned inf) const

  Check whether all index values have a certain property.
- exvector get_indices () const

  Return a vector containing the object's indices.
- exvector get_dummy_indices () const

  Return a vector containing the dummy indices of the object, if any.
- exvector get_dummy_indices (const indexed &other) const

  Return a vector containing the dummy indices in the contraction with another indexed object.
- bool has_dummy_index_for (const ex &i) const

  Check whether the object has an index that forms a dummy index pair with a given index.
- ex get_symmetry () const

  Return symmetry properties.

Public Member Functions inherited from GiNaC::container< std::vector >

- container (STLT const &s)
- container (STLT &&v)
- container (exvector::const_iterator b, exvector::const_iterator e)
- container (std::initializer_list< ex > il)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- container & prepend (const ex &b)

  Add element at front.
- container & append (const ex &b)

  Add element at back.
- container & remove_first ()

  Remove first element.
- container & remove_last ()

Remove last element.

- container & remove_all ()

  Remove all elements.

- container & sort ()

  Sort elements.

- container & unique ()

  Remove adjacent duplicate elements.

- const_iterator begin () const
- const_iterator end () const
- const_reverse_iterator rbegin () const
- const_reverse_iterator rend () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*

- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.

- virtual basic * duplicate () const

  Create a clone of this object on the heap.

- virtual ex evalf () const

  Evaluate object numerically.

- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.

- virtual ex eval_integ () const

  Evaluate integrals, if result is known.

- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.

- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.

- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.

- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.

- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.

- virtual int degree (const ex &s) const

  Return degree of highest power in object s.

- virtual int ldegree (const ex &s) const

Return degree of lowest power in object s.

- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex eval_ncmul (const exvector &v) const override

  Perform automatic simplification on noncommutative product of clifford objects.
- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- ex thiscontainer (const exvector &v) const override

  Similar to duplicate(), but with a preset sequence.
- ex thiscontainer (exvector &&v) const override

  Similar to duplicate(), but with a preset sequence (which gets pilfered).
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- void do_print_dflt (const print_dflt &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const

Protected Member Functions inherited from GiNaC::indexed

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for an indexed object always returns 0.
- ex thiscontainer (const exvector &v) const override

  Similar to duplicate(), but with a preset sequence.
- ex thiscontainer (exvector &&v) const override

  Similar to duplicate(), but with a preset sequence (which gets pilfered).
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- void printindices (const print_context &c, unsigned level) const
- void print_indexed (const print_context &c, const char *openbrace, const char *closebrace, unsigned level) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void validate () const

  Check whether all indices are of class idx and validate the symmetry tree.

Protected Member Functions inherited from GiNaC::container< std::vector >

- ex conjugate () const override
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- virtual void printseq (const print_context &c, char openbracket, char delim, char closebracket, unsigned this↩_precedence, unsigned upper_precedence=0) const

  Print sequence of contained elements.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python (const print_python &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const
- STLT subschildren (const exmap &m, unsigned options=0) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Member Functions inherited from GiNaC::container_storage< C >

- container_storage ()
- container_storage (size_t n, const ex &e)
- container_storage (std::initializer_list< ex > il)
- template<class In>
  container_storage (In b, In e)
- void reserve (size_t)
- ~container_storage ()
- void reserve (size_t n)
- void reserve (std::vector< ex > &v, size_t n)

Protected Attributes

- unsigned char representation_label

  Representation label to distinguish independent spin lines.
- ex metric

  Metric of the space, all constructors make it an indexed object.
- int commutator_sign

  It is the sign in the definition e~i e~j +/- e~j e~i = B(i, j) + B(j, i).

Protected Attributes inherited from GiNaC::indexed

- ex symtree

  Index symmetry (tree of symmetry objects).

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

Protected Attributes inherited from GiNaC::container_storage< C >

- STLT seq

Additional Inherited Members

Public Types inherited from GiNaC::container< std::vector >

- typedef STLT::const_iterator const_iterator
- typedef STLT::const_reverse_iterator const_reverse_iterator

Protected Types inherited from GiNaC::container< std::vector >

- typedef container_storage< std::vector >::STLT STLT

Protected Types inherited from GiNaC::container_storage< C >

- typedef C< ex > STLT

Static Protected Member Functions inherited from GiNaC::container< std::vector >

- static unsigned get_default_flags ()
- static char get_open_delim ()
- static char get_close_delim ()

Static Protected Member Functions inherited from GiNaC::container_storage< C >

- static void reserve (STLT &, size_t)

8.13.1 Detailed Description

This class holds an object representing an element of the Clifford algebra (the Dirac gamma matrices).

These objects only carry Lorentz indices. Spinor indices are hidden. A representation label (an unsigned 8-bit integer) is used to distinguish elements from different Clifford algebras (objects with different labels commutate).

8.13.2 Constructor & Destructor Documentation

8.13.2.1 clifford() [1/4]

```
GiNaC::clifford::clifford (
            const ex & b,
            unsigned char rl = 0)
```

Construct object without any indices.

This constructor is for internal use only. Use the dirac_ONE() function instead.

See also

dirac_ONE

References clifford(), commutator_sign, metric, and representation_label.

Referenced by clifford(), GiNaC::clifford_unit(), GiNaC::dirac_gamma(), GiNaC::dirac_gamma5(), GiNaC::dirac_gammaL(), GiNaC::dirac_gammaR(), GiNaC::dirac_ONE(), GiNaC::dirac_slash(), eval_ncmul(), match_same_type(), subs(), thiscontainer(), and thiscontainer().

8.13.2.2 clifford() [2/4]

```
GiNaC::clifford::clifford (
             const ex & b,
             const ex & mu,
             const ex & metr,
             unsigned char rl = 0,
             int comm_sign = -1)
```

Construct object with one Lorentz index.

This constructor is for internal use only. Use the clifford_unit() or dirac_gamma() functions instead.

See also

clifford_unit

dirac_gamma

References commutator_sign, GINAC_ASSERT, GiNaC::is_a(), metric, and representation_label.

8.13.2.3 clifford() [3/4]

```
GiNaC::clifford::clifford (
             unsigned char rl,
             const ex & metr,
             int comm_sign,
             const exvector & v)
```

References commutator_sign, metric, GiNaC::not_symmetric(), and representation_label.

8.13.2.4 clifford() [4/4]

```
GiNaC::clifford::clifford (
             unsigned char rl,
             const ex & metr,
             int comm_sign,
             exvector && v)
```

References commutator_sign, metric, GiNaC::not_symmetric(), and representation_label.

8.13.3 Member Function Documentation

8.13.3.1 precedence()

```
unsigned GiNaC::clifford::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

Referenced by do_print_dflt(), and do_print_latex().

8.13.3.2 archive()

```
void GiNaC::clifford::archive (
            archive_node & n) const  [override], [virtual]
```

Save (serialize) the object into archive node.

Archive the object.

Losely speaking, this method turns an expression into a byte stream (which can be saved and restored later on, or sent via network, etc.)

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_ex(), GiNaC::archive_node::add_unsigned(), commutator_sign, metric, and representation_label.

8.13.3.3 read_archive()

```
void GiNaC::clifford::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented from GiNaC::basic.

References commutator_sign, GiNaC::archive_node::find_ex(), GiNaC::archive_node::find_unsigned(), metric, and representation_label.

8.13.3.4 eval_ncmul()

```
ex GiNaC::clifford::eval_ncmul (
            const exvector & v) const  [override], [protected], [virtual]
```

Perform automatic simplification on noncommutative product of clifford objects.

This removes superfluous ONEs, permutes gamma5/L/R's to the front and removes squares of gamma objects.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::class_info< OPT >::first, clifford(), GiNaC::dirac_ONE(), GiNaC::dynallocate(), GiNaC::ex_to(), get_commutator_sign(), GiNaC::hold_ncmul(), GiNaC::indexed::indexed(), GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::ex::op(), GiNaC::reeval_ncmul(), representation_label, same_metric(), and GiNaC::varidx::toggle_variance().

8.13.3.5 match_same_type()

```
bool GiNaC::clifford::match_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::basic.

References clifford(), commutator_sign, get_commutator_sign(), GINAC_ASSERT, GiNaC::is_a(), representation_label, and same_metric().

8.13.3.6 thiscontainer() [1/2]

```
ex GiNaC::clifford::thiscontainer (
            const exvector & v) const  [override], [protected], [virtual]
```

Similar to duplicate(), but with a preset sequence.

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References clifford(), commutator_sign, metric, and representation_label.

8.13.3.7 thiscontainer() [2/2]

```
ex GiNaC::clifford::thiscontainer (
            exvector && v) const  [override], [protected], [virtual]
```

Similar to duplicate(), but with a preset sequence (which gets pilfered).

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References clifford(), commutator_sign, metric, and representation_label.

8.13.3.8 return_type()

```
unsigned GiNaC::clifford::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::noncommutative.

8.13.3.9 return_type_tinfo()

```
return_type_t GiNaC::clifford::return_type_tinfo () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::make_return_type_t(), and representation_label.

8.13.3.10 get_representation_label()

```
unsigned char GiNaC::clifford::get_representation_label () const  [inline]
```

References representation_label.

Referenced by GiNaC::cliffordunit::contract_with().

8.13.3.11 get_metric() [1/2]

```
ex GiNaC::clifford::get_metric () const  [inline]
```

References metric.

Referenced by GiNaC::cliffordunit::contract_with(), and same_metric().

8.13.3.12 get_metric() [2/2]

```
ex GiNaC::clifford::get_metric (
             const ex & i,
             const ex & j,
             bool symmetrised = false) const  [virtual]
```

References GiNaC::_ex1_2, GiNaC::ex_to(), GiNaC::ex::get_free_indices(), GiNaC::indexed::get_symmetry(), GiNaC::indexed::indexed(), GiNaC::is_a(), metric, GiNaC::subs_options::no_pattern, GiNaC::ex::op(), GiNaC::indexed::simplify_index GiNaC::ex::subs(), and GiNaC::symmetric2().

8.13.3.13 same_metric()

```
bool GiNaC::clifford::same_metric (
             const ex & other) const
```

References GiNaC::ex_to(), GiNaC::ex::get_free_indices(), get_metric(), GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::ex::is_zero(), op(), GiNaC::ex::op(), and GiNaC::indexed::simplify_indexed.

Referenced by GiNaC::cliffordunit::contract_with(), eval_ncmul(), and match_same_type().

8.13.3.14 get_commutator_sign()

```
int GiNaC::clifford::get_commutator_sign () const  [inline]
```

References commutator_sign.

Referenced by eval_ncmul(), and match_same_type().

8.13.3.15 nops()

```
size_t GiNaC::clifford::nops () const  [inline], [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

Referenced by let_op(), and op().

8.13.3.16 op()

```
ex GiNaC::clifford::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References GINAC_ASSERT, nops(), and representation_label.

Referenced by same_metric().

8.13.3.17 let_op()

```
ex & GiNaC::clifford::let_op (
            size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ensure_if_modifiable(), GINAC_ASSERT, nops(), GiNaC::numeric::numeric(), and representation_label.

8.13.3.18 subs()

```
ex GiNaC::clifford::subs (
            const exmap & m,
            unsigned options = 0) const  [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), clifford(), GiNaC::ex_to(), GiNaC::is_a(), metric, and GiNaC::ex::subs().

8.13.3.19 do_print_dflt()

```
void GiNaC::clifford::do_print_dflt (
            const print_dflt & c,
            unsigned level) const  [protected]
```

References GiNaC::is_dirac_slash(), precedence(), GiNaC::basic::print_dispatch(), GiNaC::indexed::printindices(), representation_label, GiNaC::print_context::s, and GiNaC::container_storage< C >::seq.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.13.3.20 do_print_latex()

```
void GiNaC::clifford::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

References GiNaC::is_dirac_slash(), precedence(), GiNaC::basic::print_dispatch(), representation_label, GiNaC::print_context::s, and GiNaC::container_storage< C >::seq.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.13.3.21 do_print_tree()

```
void GiNaC::clifford::do_print_tree (
            const print_tree & c,
            unsigned level) const  [protected]
```

References GiNaC::print_tree::delta_indent, GiNaC::basic::flags, GiNaC::basic::hashvalue, metric, GiNaC::ex::print(), GiNaC::indexed::printindices(), GiNaC::print_context::s, GiNaC::container_storage< C >::seq, and GiNaC::indexed::symtree.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.13.4 Member Data Documentation

8.13.4.1 representation_label

```
unsigned char GiNaC::clifford::representation_label  [protected]
```

Representation label to distinguish independent spin lines.

Referenced by archive(), clifford(), clifford(), clifford(), clifford(), do_print_dflt(), do_print_latex(), eval_ncmul(), get_representation_label(), let_op(), match_same_type(), op(), read_archive(), return_type_tinfo(), thiscontainer(), and thiscontainer().

8.13.4.2 metric

```
ex GiNaC::clifford::metric  [protected]
```

Metric of the space, all constructors make it an indexed object.

Referenced by archive(), clifford(), clifford(), clifford(), clifford(), do_print_tree(), get_metric(), get_metric(), read_archive(), subs(), thiscontainer(), and thiscontainer().

8.13.4.3 commutator_sign

```
int GiNaC::clifford::commutator_sign  [protected]
```

It is the sign in the definition e∼i e∼j +/- e∼j e∼i = B(i, j) + B(j, i).

Referenced by archive(), clifford(), clifford(), clifford(), clifford(), get_commutator_sign(), match_same_type(), read_archive(), thiscontainer(), and thiscontainer().

The documentation for this class was generated from the following files:

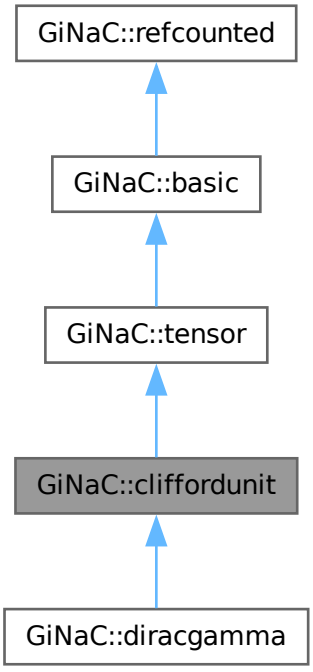
- clifford.h
- clifford.cpp

8.14 GiNaC::cliffordunit Class Reference

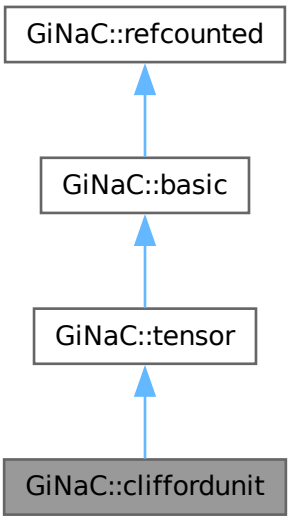
This class represents the Clifford algebra generators (units).

```
#include <clifford.h>
```

Inheritance diagram for GiNaC::cliffordunit:



Collaboration diagram for GiNaC::cliffordunit:



Public Member Functions

- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of a Clifford unit with something else.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.

- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().

- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.14.1 Detailed Description

This class represents the Clifford algebra generators (units).

8.14.2 Member Function Documentation

8.14.2.1 contract_with()

```
bool GiNaC::cliffordunit::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [override], [virtual]
```

Contraction of a Clifford unit with something else.

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::diracgamma.

References GiNaC::_ex1, GiNaC::dirac_ONE(), GiNaC::ex_to(), GiNaC::clifford::get_metric(), GiNaC::clifford::get_representation_lab
GINAC_ASSERT, GiNaC::is_a(), GiNaC::ncmul::ncmul(), GiNaC::ex::op(), and GiNaC::clifford::same_metric().

8.14.2.2 do_print()

```
void GiNaC::cliffordunit::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

8.14.2.3 do_print_latex()

```
void GiNaC::cliffordunit::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

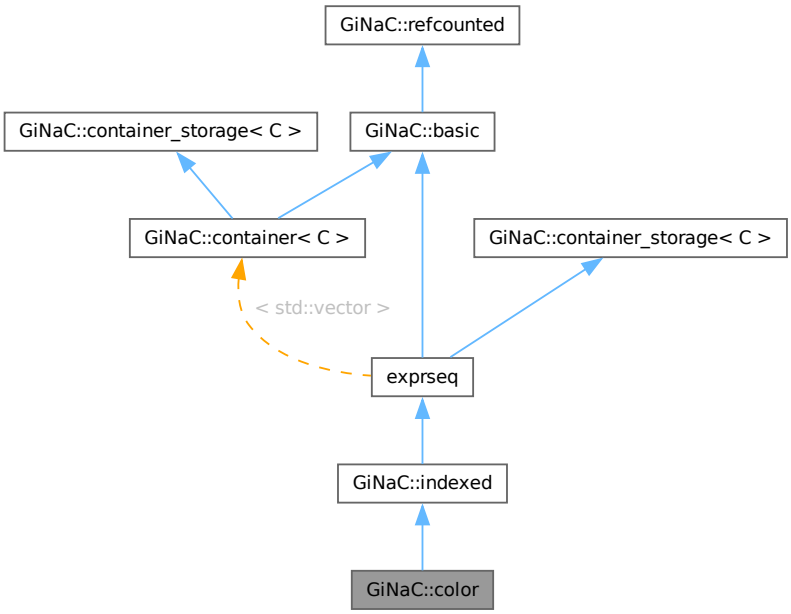
- clifford.h
- clifford.cpp

8.15 GiNaC::color Class Reference

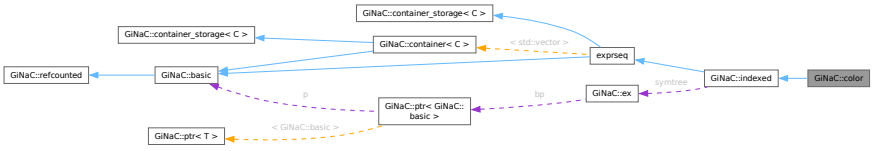
This class holds a generator T_a or the unity element of the Lie algebra of SU(3), as used for calculations in quantum chromodynamics.

```
#include <color.h>
```

Inheritance diagram for GiNaC::color:



Collaboration diagram for GiNaC::color:



Public Member Functions

- color (const ex &b, unsigned char rl=0)

  Construct object without any color index.
- color (const ex &b, const ex &i1, unsigned char rl=0)

  Construct object with one color index.
- color (unsigned char rl, const exvector &v)
- color (unsigned char rl, exvector &&v)
- void archive (archive_node &n) const override

  Save (serialize) the object into archive node.
- void read_archive (const archive_node &n, lst &sym_lst) override

  Load (deserialize) the object from an archive node.
- unsigned char get_representation_label () const

Public Member Functions inherited from GiNaC::indexed

- indexed (const ex &b)

  Construct indexed object with no index.
- indexed (const ex &b, const ex &i1)

  Construct indexed object with one index.
- indexed (const ex &b, const ex &i1, const ex &i2)

  Construct indexed object with two indices.
- indexed (const ex &b, const ex &i1, const ex &i2, const ex &i3)

  Construct indexed object with three indices.
- indexed (const ex &b, const ex &i1, const ex &i2, const ex &i3, const ex &i4)

  Construct indexed object with four indices.
- indexed (const ex &b, const symmetry &symm, const ex &i1, const ex &i2)

  Construct indexed object with two indices and a specified symmetry.
- indexed (const ex &b, const symmetry &symm, const ex &i1, const ex &i2, const ex &i3)

  Construct indexed object with three indices and a specified symmetry.
- indexed (const ex &b, const symmetry &symm, const ex &i1, const ex &i2, const ex &i3, const ex &i4)

  Construct indexed object with four indices and a specified symmetry.
- indexed (const ex &b, const exvector &iv)

  Construct indexed object with a specified vector of indices.
- indexed (const ex &b, const symmetry &symm, const exvector &iv)

  Construct indexed object with a specified vector of indices and symmetry.
- indexed (const symmetry &symm, const exprseq &es)
- indexed (const symmetry &symm, const exvector &v)
- indexed (const symmetry &symm, exvector &&v)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex real_part () const override
- ex imag_part () const override
- exvector get_free_indices () const override

  Return a vector containing the free indices of an expression.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- bool all_index_values_are (unsigned inf) const

  Check whether all index values have a certain property.
- exvector get_indices () const

  Return a vector containing the object's indices.
- exvector get_dummy_indices () const

  Return a vector containing the dummy indices of the object, if any.
- exvector get_dummy_indices (const indexed &other) const

  Return a vector containing the dummy indices in the contraction with another indexed object.
- bool has_dummy_index_for (const ex &i) const

  Check whether the object has an index that forms a dummy index pair with a given index.
- ex get_symmetry () const

  Return symmetry properties.

Public Member Functions inherited from GiNaC::container< std::vector >

- container (STLT const &s)
- container (STLT &&v)
- container (exvector::const_iterator b, exvector::const_iterator e)
- container (std::initializer_list< ex > il)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- container & prepend (const ex &b)

  Add element at front.
- container & append (const ex &b)

  Add element at back.
- container & remove_first ()

  Remove first element.
- container & remove_last ()

  Remove last element.
- container & remove_all ()

  Remove all elements.
- container & sort ()

  Sort elements.
- container & unique ()

  Remove adjacent duplicate elements.
- const_iterator begin () const
- const_iterator end () const
- const_reverse_iterator rbegin () const
- const_reverse_iterator rend () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

Output to stream.

- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().

- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex eval_ncmul (const exvector &v) const override

  Perform automatic simplification on noncommutative product of color objects.
- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- ex thiscontainer (const exvector &v) const override

  Similar to duplicate(), but with a preset sequence.
- ex thiscontainer (exvector &&v) const override

  Similar to duplicate(), but with a preset sequence (which gets pilfered).
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override

Protected Member Functions inherited from GiNaC::indexed

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for an indexed object always returns 0.
- ex thiscontainer (const exvector &v) const override

  Similar to duplicate(), but with a preset sequence.
- ex thiscontainer (exvector &&v) const override

  Similar to duplicate(), but with a preset sequence (which gets pilfered).
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- void printindices (const print_context &c, unsigned level) const
- void print_indexed (const print_context &c, const char ∗openbrace, const char ∗closebrace, unsigned level) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void validate () const

  Check whether all indices are of class idx and validate the symmetry tree.

Protected Member Functions inherited from GiNaC::container< std::vector >

- ex conjugate () const override
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- virtual void printseq (const print_context &c, char openbracket, char delim, char closebracket, unsigned this_precedence, unsigned upper_precedence=0) const

  Print sequence of contained elements.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python (const print_python &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const
- STLT subschildren (const exmap &m, unsigned options=0) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Member Functions inherited from GiNaC::container_storage< C >

- container_storage ()
- container_storage (size_t n, const ex &e)
- container_storage (std::initializer_list< ex > il)
- template<class In>
  container_storage (In b, In e)
- void reserve (size_t)
- ~container_storage ()
- void reserve (size_t n)
- void reserve (std::vector< ex > &v, size_t n)

Private Attributes

- unsigned char representation_label

  Representation label to distinguish independent color matrices coming from separated fermion lines.

Additional Inherited Members

Public Types inherited from GiNaC::container< std::vector >

- typedef STLT::const_iterator const_iterator
- typedef STLT::const_reverse_iterator const_reverse_iterator

Protected Types inherited from GiNaC::container< std::vector >

- typedef container_storage< std::vector >::STLT STLT

Protected Types inherited from GiNaC::container_storage< C >

- typedef C< ex > STLT

Static Protected Member Functions inherited from GiNaC::container< std::vector >

- static unsigned get_default_flags ()
- static char get_open_delim ()
- static char get_close_delim ()

Static Protected Member Functions inherited from GiNaC::container_storage< C >

- static void reserve (STLT &, size_t)

Protected Attributes inherited from GiNaC::indexed

- ex symtree

  Index symmetry (tree of symmetry objects).

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

Protected Attributes inherited from GiNaC::container_storage< C >

- STLT seq

8.15.1 Detailed Description

This class holds a generator T_a or the unity element of the Lie algebra of SU(3), as used for calculations in quantum chromodynamics.

A representation label (an unsigned 8-bit integer) is used to distinguish elements from different Lie algebras (objects with different labels commutate). These objects implement an abstract representation of the group, not a specific matrix representation. The indices used for color objects should not have a variance.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 color() [1/4]

```
GiNaC::color::color (
             const ex & b,
             unsigned char rl = 0)
```

Construct object without any color index.

This constructor is for internal use only. Use the color_ONE() function instead.

See also

color_ONE

References color(), and representation_label.

Referenced by color(), GiNaC::color_ONE(), GiNaC::color_T(), eval_ncmul(), match_same_type(), thiscontainer(), and thiscontainer().

8.15.2.2 color() [2/4]

```
GiNaC::color::color (
             const ex & b,
             const ex & i1,
             unsigned char rl = 0)
```

Construct object with one color index.

This constructor is for internal use only. Use the color_T() function instead.

See also

color_T

References representation_label.

8.15.2.3 color() [3/4]

```
GiNaC::color::color (
            unsigned char rl,
            const exvector & v)
```

References GiNaC::not_symmetric(), and representation_label.

8.15.2.4 color() [4/4]

```
GiNaC::color::color (
            unsigned char rl,
            exvector && v)
```

References GiNaC::not_symmetric(), and representation_label.

8.15.3 Member Function Documentation

8.15.3.1 archive()

```
void GiNaC::color::archive (
            archive_node & n) const  [override], [virtual]
```

Save (serialize) the object into archive node.

Archive the object.

Losely speaking, this method turns an expression into a byte stream (which can be saved and restored later on, or sent via network, etc.)

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_unsigned(), and representation_label.

8.15.3.2 read_archive()

```
void GiNaC::color::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::find_unsigned(), and representation_label.

8.15.3.3 eval_ncmul()

```
ex GiNaC::color::eval_ncmul (
            const exvector & v) const  [override], [protected], [virtual]
```

Perform automatic simplification on noncommutative product of color objects.

This removes superfluous ONEs.

Reimplemented from GiNaC::basic.

References color(), eval_ncmul(), GiNaC::hold_ncmul(), GiNaC::is_a(), and representation_label.

Referenced by eval_ncmul().

8.15.3.4 match_same_type()

```
bool GiNaC::color::match_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::basic.

References color(), GINAC_ASSERT, GiNaC::is_a(), and representation_label.

8.15.3.5 thiscontainer() [1/2]

```
ex GiNaC::color::thiscontainer (
            const exvector & v) const  [override], [protected], [virtual]
```

Similar to duplicate(), but with a preset sequence.

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References color(), and representation_label.

8.15.3.6 thiscontainer() [2/2]

```
ex GiNaC::color::thiscontainer (
            exvector && v) const  [override], [protected], [virtual]
```

Similar to duplicate(), but with a preset sequence (which gets pilfered).

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References color(), and representation_label.

8.15.3.7 return_type()

```
unsigned GiNaC::color::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::noncommutative.

8.15.3.8 return_type_tinfo()

```
return_type_t GiNaC::color::return_type_tinfo () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::make_return_type_t(), and representation_label.

8.15.3.9 get_representation_label()

```
unsigned char GiNaC::color::get_representation_label () const  [inline]
```

References representation_label.

8.15.4 Member Data Documentation

8.15.4.1 representation_label

```
unsigned char GiNaC::color::representation_label  [private]
```

Representation label to distinguish independent color matrices coming from separated fermion lines.

Referenced by archive(), color(), color(), color(), color(), eval_ncmul(), get_representation_label(), match_same_type(), read_archive(), return_type_tinfo(), thiscontainer(), and thiscontainer().

The documentation for this class was generated from the following files:

- color.h
- color.cpp

8.16 GiNaC::compare_all_equal< T > Class Template Reference

Comparison policy: all structures of one type are equal.

`#include <structure.h>`

Protected Member Functions

- ~compare_all_equal ()

Static Protected Member Functions

- static bool struct_is_equal (const T *t1, const T *t2)
- static int struct_compare (const T *t1, const T *t2)

8.16.1 Detailed Description

template<class T>
class GiNaC::compare_all_equal< T >

Comparison policy: all structures of one type are equal.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 ~compare_all_equal()

```
template<class T>
GiNaC::compare_all_equal< T >::~compare_all_equal ()  [inline], [protected]
```

8.16.3 Member Function Documentation

8.16.3.1 struct_is_equal()

```
template<class T>
bool GiNaC::compare_all_equal< T >::struct_is_equal (
            const T * t1,
            const T * t2)  [inline], [static], [protected]
```

8.16.3.2 struct_compare()

```
template<class T>
int GiNaC::compare_all_equal< T >::struct_compare (
            const T * t1,
            const T * t2)  [inline], [static], [protected]
```

The documentation for this class was generated from the following file:

- structure.h

8.17 GiNaC::compare_bitwise< T > Class Template Reference

Comparison policy: use bit-wise comparison to compare structures.

`#include <structure.h>`

Protected Member Functions

- ~compare_bitwise ()

Static Protected Member Functions

- static bool struct_is_equal (const T *t1, const T *t2)
- static int struct_compare (const T *t1, const T *t2)

8.17.1 Detailed Description

template<class T>
class GiNaC::compare_bitwise< T >

Comparison policy: use bit-wise comparison to compare structures.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 ~compare_bitwise()

```
template<class T>
GiNaC::compare_bitwise< T >::~compare_bitwise ()  [inline], [protected]
```

8.17.3 Member Function Documentation

8.17.3.1 struct_is_equal()

```
template<class T>
bool GiNaC::compare_bitwise< T >::struct_is_equal (
            const T * t1,
            const T * t2)  [inline], [static], [protected]
```

8.17.3.2 struct_compare()

```
template<class T>
int GiNaC::compare_bitwise< T >::struct_compare (
            const T * t1,
            const T * t2)  [inline], [static], [protected]
```

The documentation for this class was generated from the following file:

- structure.h

8.18 GiNaC::compare_std_less< T > Class Template Reference

Comparison policy: use std::equal_to/std::less (defaults to operators == and <) to compare structures.

`#include <structure.h>`

Protected Member Functions

- ~compare_std_less ()

Static Protected Member Functions

- static bool struct_is_equal (const T *t1, const T *t2)
- static int struct_compare (const T *t1, const T *t2)

8.18.1 Detailed Description

template<class T>
class GiNaC::compare_std_less< T >

Comparison policy: use std::equal_to/std::less (defaults to operators == and <) to compare structures.

8.18.2 Constructor & Destructor Documentation

8.18.2.1 ~compare_std_less()

```
template<class T>
GiNaC::compare_std_less< T >::~compare_std_less ()  [inline], [protected]
```

8.18.3 Member Function Documentation

8.18.3.1 struct_is_equal()

```
template<class T>
bool GiNaC::compare_std_less< T >::struct_is_equal (
            const T * t1,
            const T * t2)  [inline], [static], [protected]
```

8.18.3.2 struct_compare()

```
template<class T>
int GiNaC::compare_std_less< T >::struct_compare (
            const T * t1,
            const T * t2)  [inline], [static], [protected]
```

The documentation for this class was generated from the following file:

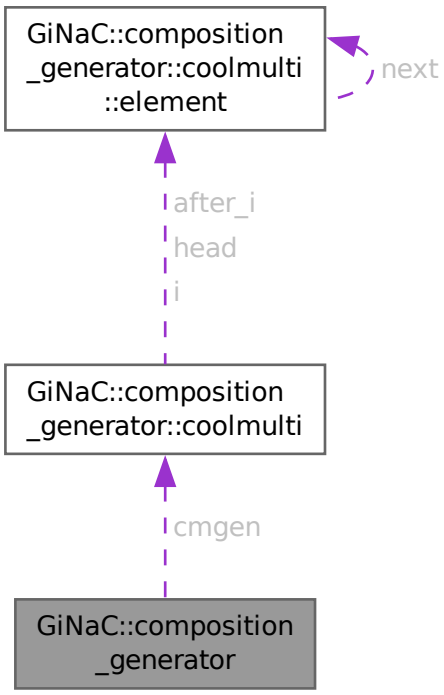
- structure.h

8.19 GiNaC::composition_generator Class Reference

Generate all compositions of a partition of an integer n, starting with the compositions which has non-decreasing order.

`#include <utils.h>`

Collaboration diagram for GiNaC::composition_generator:



Classes

- struct coolmulti

Public Member Functions

- composition_generator (const std::vector< unsigned > &partition)
- const std::vector< unsigned > & get () const
- bool next ()

Private Attributes

- struct GiNaC::composition_generator::coolmulti cmgen
- bool atend
- bool trivial
- std::vector< unsigned > composition
- bool current_updated

8.19.1 Detailed Description

Generate all compositions of a partition of an integer n, starting with the compositions which has non-decreasing order.

8.19.2 Constructor & Destructor Documentation

8.19.2.1 composition_generator()

```
GiNaC::composition_generator::composition_generator (
            const std::vector< unsigned > & partition)  [inline], [explicit]
```

References atend, cmgen, composition, current_updated, and trivial.

8.19.3 Member Function Documentation

8.19.3.1 get()

```
const std::vector< unsigned > & GiNaC::composition_generator::get () const  [inline]
```

References cmgen, composition, current_updated, GiNaC::composition_generator::coolmulti::head, GiNaC::composition_generator::c and GiNaC::composition_generator::coolmulti::element::value.

Referenced by GiNaC::power::expand_add().

8.19.3.2 next()

```
bool GiNaC::composition_generator::next ()  [inline]
```

References atend, cmgen, current_updated, GiNaC::composition_generator::coolmulti::finished(), GiNaC::composition_generator::co and trivial.

Referenced by GiNaC::power::expand_add().

8.19.4 Member Data Documentation

8.19.4.1 cmgen

```
struct GiNaC::composition_generator::coolmulti GiNaC::composition_generator::cmgen  [private]
```

Referenced by composition_generator(), get(), and next().

8.19.4.2 atend

```
bool GiNaC::composition_generator::atend  [private]
```

Referenced by composition_generator(), and next().

8.19.4.3 trivial

```
bool GiNaC::composition_generator::trivial  [private]
```

Referenced by composition_generator(), and next().

8.19.4.4 composition

```
std::vector<unsigned> GiNaC::composition_generator::composition  [mutable], [private]
```

Referenced by composition_generator(), and get().

8.19.4.5 current_updated

```
bool GiNaC::composition_generator::current_updated  [mutable], [private]
```

Referenced by composition_generator(), get(), and next().

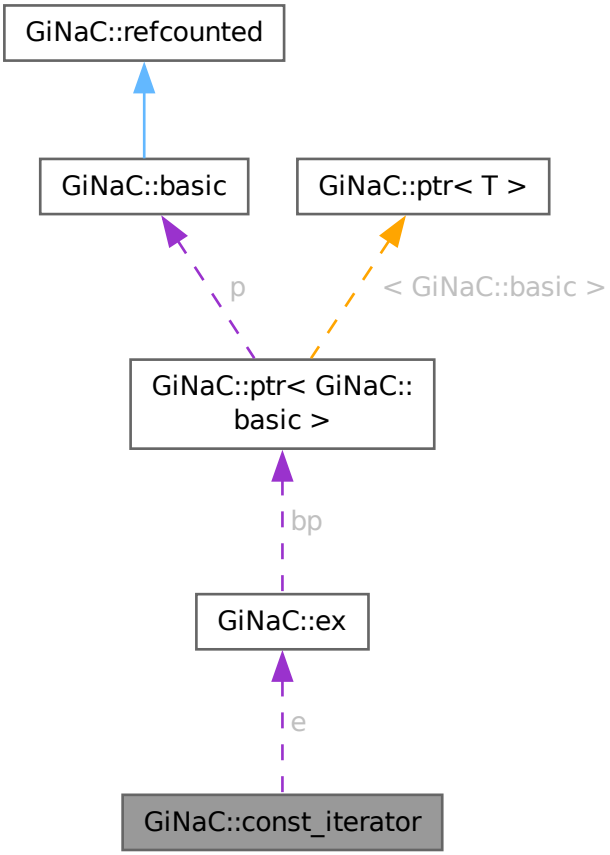
The documentation for this class was generated from the following file:

- utils.h

8.20 GiNaC::const_iterator Class Reference

```
#include <ex.h>
```

Collaboration diagram for GiNaC::const_iterator:



Public Types

- using iterator_category = std::random_access_iterator_tag
- using value_type = ex
- using difference_type = ptrdiff_t
- using pointer = const ex *
- using reference = const ex &

Public Member Functions

- const_iterator () noexcept
- ex operator* () const
- std::unique_ptr< ex > operator-> () const
- ex operator[ ] (difference_type n) const
- const_iterator & operator++ () noexcept
- const_iterator operator++ (int) noexcept
- const_iterator & operator+= (difference_type n) noexcept
- const_iterator operator+ (difference_type n) const noexcept

- const_iterator & operator-- () noexcept
- const_iterator operator-- (int) noexcept
- const_iterator & operator-= (difference_type n) noexcept
- const_iterator operator- (difference_type n) const noexcept
- bool operator== (const const_iterator &other) const noexcept
- bool operator!= (const const_iterator &other) const noexcept
- bool operator< (const const_iterator &other) const noexcept
- bool operator> (const const_iterator &other) const noexcept
- bool operator<= (const const_iterator &other) const noexcept
- bool operator>= (const const_iterator &other) const noexcept

Protected Attributes

- ex e
- size_t i

Private Member Functions

- const_iterator (const ex &e_, size_t i_) noexcept

Friends

- class ex
- class const_preorder_iterator
- class const_postorder_iterator
- const_iterator operator+ (difference_type n, const const_iterator &it) noexcept
- difference_type operator- (const const_iterator &lhs, const const_iterator &rhs) noexcept

8.20.1 Member Typedef Documentation

8.20.1.1 iterator_category

```
using GiNaC::const_iterator::iterator_category = std::random_access_iterator_tag
```

8.20.1.2 value_type

```
using GiNaC::const_iterator::value_type = ex
```

8.20.1.3 difference_type

```
using GiNaC::const_iterator::difference_type = ptrdiff_t
```

8.20.1.4 pointer

```
using GiNaC::const_iterator::pointer = const ex *
```

8.20.1.5 reference

```
using GiNaC::const_iterator::reference = const ex &
```

8.20.2 Constructor & Destructor Documentation

8.20.2.1 const_iterator() [1/2]

```
GiNaC::const_iterator::const_iterator ()  [inline], [noexcept]
```

Referenced by GiNaC::ex::begin(), GiNaC::ex::end(), operator!=(), operator+(), operator+, operator++(), operator++(), operator+=(), operator-, operator-(), operator--(), operator--(), operator-=(), operator<(), operator<=(), operator==(), operator>(), and operator>=().

8.20.2.2 const_iterator() [2/2]

```
GiNaC::const_iterator::const_iterator (
            const ex & e_,
            size_t i_)  [inline], [private], [noexcept]
```

References e, ex, and i.

8.20.3 Member Function Documentation

8.20.3.1 operator*()

```
ex GiNaC::const_iterator::operator* () const  [inline]
```

References e, ex, i, and GiNaC::ex::op().

8.20.3.2 operator->()

```
std::unique_ptr< ex > GiNaC::const_iterator::operator-> () const  [inline]
```

References GiNaC::ex::ex().

8.20.3.3 operator[]()

```
ex GiNaC::const_iterator::operator[] (
            difference_type n) const  [inline]
```

References e, ex, i, and GiNaC::ex::op().

8.20.3.4 operator++() [1/2]

```
const_iterator & GiNaC::const_iterator::operator++ ()  [inline], [noexcept]
```

References const_iterator(), and i.

8.20.3.5 operator++() [2/2]

```
const_iterator GiNaC::const_iterator::operator++ (
            int )  [inline], [noexcept]
```

References const_iterator(), and i.

8.20.3.6 operator+=()

```
const_iterator & GiNaC::const_iterator::operator+= (
            difference_type n)  [inline], [noexcept]
```

References const_iterator(), and i.

8.20.3.7 operator+()

```
const_iterator GiNaC::const_iterator::operator+ (
            difference_type n) const  [inline], [noexcept]
```

References const_iterator(), e, and i.

8.20.3.8 operator--() [1/2]

```
const_iterator & GiNaC::const_iterator::operator-- ()  [inline], [noexcept]
```

References const_iterator(), and i.

8.20.3.9 operator--() [2/2]

```
const_iterator GiNaC::const_iterator::operator-- (
            int )  [inline], [noexcept]
```

References const_iterator(), and i.

8.20.3.10 operator-=()

```
const_iterator & GiNaC::const_iterator::operator-= (
            difference_type n)  [inline], [noexcept]
```

References const_iterator(), and i.

8.20.3.11 operator-()

```
const_iterator GiNaC::const_iterator::operator- (
            difference_type n) const  [inline], [noexcept]
```

References const_iterator(), e, and i.

8.20.3.12 operator==()

```
bool GiNaC::const_iterator::operator== (
            const const_iterator & other) const  [inline], [noexcept]
```

References GiNaC::are_ex_trivially_equal(), const_iterator(), e, and i.

8.20.3.13 operator"!=()

```
bool GiNaC::const_iterator::operator!= (
            const const_iterator & other) const  [inline], [noexcept]
```

References const_iterator().

8.20.3.14 operator<()

```
bool GiNaC::const_iterator::operator< (
            const const_iterator & other) const  [inline], [noexcept]
```

References const_iterator(), and i.

8.20.3.15 operator>()

```
bool GiNaC::const_iterator::operator> (
            const const_iterator & other) const  [inline], [noexcept]
```

References const_iterator().

8.20.3.16 operator<=()

```
bool GiNaC::const_iterator::operator<= (
            const const_iterator & other) const  [inline], [noexcept]
```

References const_iterator().

8.20.3.17 operator>=()

```
bool GiNaC::const_iterator::operator>= (
            const const_iterator & other) const  [inline], [noexcept]
```

References const_iterator().

8.20.4 Friends And Related Symbol Documentation

8.20.4.1 ex

```
friend class ex  [friend]
```

References ex.

Referenced by const_iterator(), ex, operator*(), and operator[]().

8.20.4.2 const_preorder_iterator

```
friend class const_preorder_iterator  [friend]
```

References const_preorder_iterator.

Referenced by const_preorder_iterator.

8.20.4.3 const_postorder_iterator

```
friend class const_postorder_iterator  [friend]
```

References const_postorder_iterator.

Referenced by const_postorder_iterator.

8.20.4.4 operator+

```
const_iterator operator+ (
            difference_type n,
            const const_iterator & it)  [friend]
```

References const_iterator().

8.20.4.5 operator-

```
difference_type operator- (
            const const_iterator & lhs,
            const const_iterator & rhs)  [friend]
```

References const_iterator(), GiNaC::lhs(), and GiNaC::rhs().

8.20.5 Member Data Documentation

8.20.5.1 e

```
ex GiNaC::const_iterator::e  [protected]
```

Referenced by const_iterator(), operator∗(), operator+(), operator-(), operator==(), and operator[ ]().

8.20.5.2 i

```
size_t GiNaC::const_iterator::i  [protected]
```

Referenced by const_iterator(), operator∗(), operator+(), operator++(), operator++(), operator+=(), operator-(), operator--(), operator--(), operator-=(), operator<(), operator==(), and operator[ ]().

The documentation for this class was generated from the following file:

- ex.h

8.21 GiNaC::const_postorder_iterator Class Reference

`#include <ex.h>`

Public Types

- using iterator_category = std::forward_iterator_tag
- using value_type = ex
- using difference_type = ptrdiff_t
- using pointer = const ex *
- using reference = const ex &

Public Member Functions

- const_postorder_iterator () noexcept
- const_postorder_iterator (const ex &e, size_t n)
- reference operator* () const
- pointer operator-> () const
- const_postorder_iterator & operator++ ()
- const_postorder_iterator operator++ (int)
- bool operator== (const const_postorder_iterator &other) const noexcept
- bool operator!= (const const_postorder_iterator &other) const noexcept

Private Member Functions

- void descend ()
- void increment ()

Private Attributes

- std::stack< internal::_iter_rep, std::vector< internal::_iter_rep > > s

8.21.1 Member Typedef Documentation

8.21.1.1 iterator_category

`using GiNaC::const_postorder_iterator::iterator_category = std::forward_iterator_tag`

8.21.1.2 value_type

`using GiNaC::const_postorder_iterator::value_type = ex`

8.21.1.3 difference_type

`using GiNaC::const_postorder_iterator::difference_type = ptrdiff_t`

8.21.1.4 pointer

```
using GiNaC::const_postorder_iterator::pointer = const ex *
```

8.21.1.5 reference

```
using GiNaC::const_postorder_iterator::reference = const ex &
```

8.21.2 Constructor & Destructor Documentation

8.21.2.1 const_postorder_iterator() [1/2]

```
GiNaC::const_postorder_iterator::const_postorder_iterator ()  [inline], [noexcept]
```

Referenced by operator!=(), operator++(), operator++(), operator==(), GiNaC::ex::postorder_begin(), and GiNaC::ex::postorder_end().

8.21.2.2 const_postorder_iterator() [2/2]

```
GiNaC::const_postorder_iterator::const_postorder_iterator (
            const ex & e,
            size_t n)  [inline]
```

References descend(), and s.

8.21.3 Member Function Documentation

8.21.3.1 operator*()

```
reference GiNaC::const_postorder_iterator::operator* () const  [inline]
```

References s.

8.21.3.2 operator->()

```
pointer GiNaC::const_postorder_iterator::operator-> () const  [inline]
```

References s.

8.21.3.3 operator++() [1/2]

```
const_postorder_iterator & GiNaC::const_postorder_iterator::operator++ ()  [inline]
```

References const_postorder_iterator(), and increment().

8.21.3.4 operator++() [2/2]

```
const_postorder_iterator GiNaC::const_postorder_iterator::operator++ (
            int )  [inline]
```

References const_postorder_iterator(), and increment().

8.21.3.5 operator==()

```
bool GiNaC::const_postorder_iterator::operator== (
            const const_postorder_iterator & other) const  [inline], [noexcept]
```

References const_postorder_iterator(), and s.

8.21.3.6 operator"!=()

```
bool GiNaC::const_postorder_iterator::operator!= (
            const const_postorder_iterator & other) const  [inline], [noexcept]
```

References const_postorder_iterator().

8.21.3.7 descend()

```
void GiNaC::const_postorder_iterator::descend ()  [inline], [private]
```

References GiNaC::internal::_iter_rep::e, GiNaC::internal::_iter_rep::i, GiNaC::ex::nops(), GiNaC::ex::op(), and s.

Referenced by const_postorder_iterator(), and increment().

8.21.3.8 increment()

```
void GiNaC::const_postorder_iterator::increment ()  [inline], [private]
```

References descend(), and s.

Referenced by operator++(), and operator++().

8.21.4 Member Data Documentation

8.21.4.1 s

```
std::stack<internal::_iter_rep, std::vector<internal::_iter_rep> > GiNaC::const_postorder_↩
iterator::s  [private]
```

Referenced by const_postorder_iterator(), descend(), increment(), operator∗(), operator->(), and operator==().

The documentation for this class was generated from the following file:

- ex.h

8.22 GiNaC::const_preorder_iterator Class Reference

`#include <ex.h>`

Public Types

- using iterator_category = std::forward_iterator_tag
- using value_type = ex
- using difference_type = ptrdiff_t
- using pointer = const ex *
- using reference = const ex &

Public Member Functions

- const_preorder_iterator () noexcept
- const_preorder_iterator (const ex &e, size_t n)
- reference operator* () const
- pointer operator-> () const
- const_preorder_iterator & operator++ ()
- const_preorder_iterator operator++ (int)
- bool operator== (const const_preorder_iterator &other) const noexcept
- bool operator!= (const const_preorder_iterator &other) const noexcept

Private Member Functions

- void increment ()

Private Attributes

- std::stack< internal::_iter_rep, std::vector< internal::_iter_rep > > s

8.22.1 Member Typedef Documentation

8.22.1.1 iterator_category

`using GiNaC::const_preorder_iterator::iterator_category = std::forward_iterator_tag`

8.22.1.2 value_type

`using GiNaC::const_preorder_iterator::value_type = ex`

8.22.1.3 difference_type

`using GiNaC::const_preorder_iterator::difference_type = ptrdiff_t`

8.22.1.4 pointer

```
using GiNaC::const_preorder_iterator::pointer = const ex *
```

8.22.1.5 reference

```
using GiNaC::const_preorder_iterator::reference = const ex &
```

8.22.2 Constructor & Destructor Documentation

8.22.2.1 const_preorder_iterator() [1/2]

```
GiNaC::const_preorder_iterator::const_preorder_iterator ()  [inline], [noexcept]
```

Referenced by operator!=(), operator++(), operator++(), operator==(), GiNaC::ex::preorder_begin(), and GiNaC::ex::preorder_end().

8.22.2.2 const_preorder_iterator() [2/2]

```
GiNaC::const_preorder_iterator::const_preorder_iterator (
            const ex & e,
            size_t n)  [inline]
```

References s.

8.22.3 Member Function Documentation

8.22.3.1 operator*()

```
reference GiNaC::const_preorder_iterator::operator* () const  [inline]
```

References s.

8.22.3.2 operator->()

```
pointer GiNaC::const_preorder_iterator::operator-> () const  [inline]
```

References s.

8.22.3.3 operator++() [1/2]

```
const_preorder_iterator & GiNaC::const_preorder_iterator::operator++ ()  [inline]
```

References const_preorder_iterator(), and increment().

8.22.3.4 operator++() [2/2]

```
const_preorder_iterator GiNaC::const_preorder_iterator::operator++ (
            int )  [inline]
```

References const_preorder_iterator(), and increment().

8.22.3.5 operator==()

```
bool GiNaC::const_preorder_iterator::operator== (
            const const_preorder_iterator & other) const  [inline], [noexcept]
```

References const_preorder_iterator(), and s.

8.22.3.6 operator"!=()

```
bool GiNaC::const_preorder_iterator::operator!= (
            const const_preorder_iterator & other) const  [inline], [noexcept]
```

References const_preorder_iterator().

8.22.3.7 increment()

```
void GiNaC::const_preorder_iterator::increment ()  [inline], [private]
```

References GiNaC::internal::_iter_rep::e, GiNaC::internal::_iter_rep::i, GiNaC::internal::_iter_rep::i_end, GiNaC::ex::nops(), GiNaC::ex::op(), and s.

Referenced by operator++(), and operator++().

8.22.4 Member Data Documentation

8.22.4.1 s

```
std::stack<internal::_iter_rep, std::vector<internal::_iter_rep> > GiNaC::const_preorder_↩
iterator::s  [private]
```

Referenced by const_preorder_iterator(), increment(), operator∗(), operator->(), and operator==().

The documentation for this class was generated from the following file:

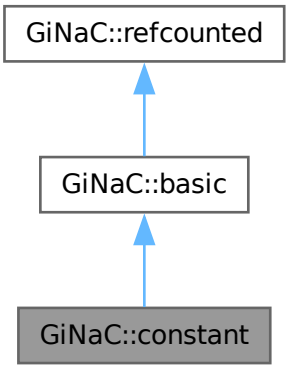
- ex.h

8.23 GiNaC::constant Class Reference

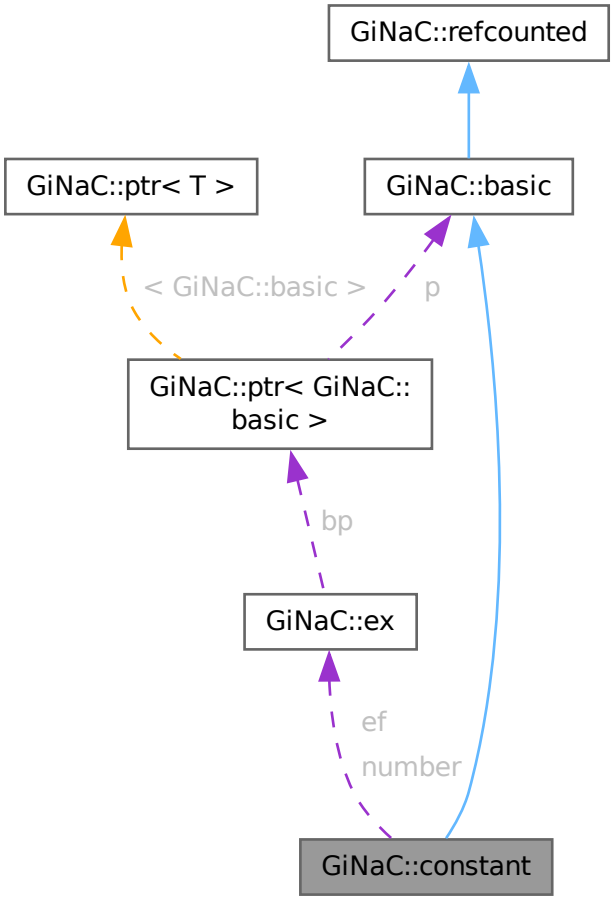
This class holds constants, symbols with specific numerical value.

`#include <constant.h>`

Inheritance diagram for GiNaC::constant:



Collaboration diagram for GiNaC::constant:



Public Member Functions

- constant (const std::string &initname, evalffunctype efun=nullptr, const std::string &texname=std::string(), unsigned domain=domain::complex)
- constant (const std::string &initname, const numeric &initnumber, const std::string &texname=std::string(), unsigned domain=domain::complex)
- bool info (unsigned inf) const override

  Information about the object.
- ex evalf () const override

  Evaluate object numerically.
- bool is_polynomial (const ex &var) const override

  Check whether this is a polynomial in the given variables.
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- void archive (archive_node &n) const override

  Save (serialize) the object into archive node.

- void read_archive (const archive_node &n, lst &syms) override

  Load (deserialize) the object from an archive node.
- std::string get_name () const
- std::string get_TeX_name () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual int degree (const ex &s) const

Return degree of highest power in object s.

- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for a constant always returns 0.
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Attributes

- std::string name

  printname of this constant
- std::string TeX_name

  LaTeX name.
- evalffunctype ef
- ex number

  numerical value this constant evalf()s to
- unsigned serial

  unique serial number for comparison
- unsigned domain

  numerical value this constant evalf()s to

Static Private Attributes

- static unsigned next_serial = 0

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags
  of type status_flags
- unsigned hashvalue
  hash value

8.23.1 Detailed Description

This class holds constants, symbols with specific numerical value.

Each object of this class must either provide their own function to evaluate it to class numeric or provide the constant as a numeric (if it's an exact number).

8.23.2 Constructor & Destructor Documentation

8.23.2.1 constant() [1/2]

```
GiNaC::constant::constant (
            const std::string & initname,
            evalffunctype efun = nullptr,
            const std::string & texname = std::string(),
            unsigned domain = domain::complex)
```

References constant(), domain, ef, GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, name, next_serial, serial, GiNaC::basic::setflag(), and TeX_name.

Referenced by constant(), and is_equal_same_type().

8.23.2.2 constant() [2/2]

```
GiNaC::constant::constant (
            const std::string & initname,
            const numeric & initnumber,
            const std::string & texname = std::string(),
            unsigned domain = domain::complex)
```

References domain, ef, GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, name, next_serial, number, serial, GiNaC::basic::setflag(), and TeX_name.

8.23.3 Member Function Documentation

8.23.3.1 info()

```
bool GiNaC::constant::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References domain, GiNaC::info_flags::nonnegative, GiNaC::info_flags::polynomial, GiNaC::domain::positive, GiNaC::info_flags::positive, GiNaC::domain::real, and GiNaC::info_flags::real.

8.23.3.2 evalf()

```
ex GiNaC::constant::evalf () const  [override], [virtual]
```

Evaluate object numerically.

Reimplemented from GiNaC::basic.

References ef, GiNaC::ex::evalf(), GiNaC::basic::ex, and number.

8.23.3.3 is_polynomial()

```
bool GiNaC::constant::is_polynomial (
            const ex & var) const  [override], [virtual]
```

Check whether this is a polynomial in the given variables.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.23.3.4 conjugate()

```
ex GiNaC::constant::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References domain, GiNaC::basic::ex, GiNaC::domain::positive, and GiNaC::domain::real.

8.23.3.5 real_part()

```
ex GiNaC::constant::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References domain, GiNaC::basic::ex, GiNaC::domain::positive, and GiNaC::domain::real.

8.23.3.6 imag_part()

```
ex GiNaC::constant::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References domain, GiNaC::basic::ex, GiNaC::domain::positive, and GiNaC::domain::real.

8.23.3.7 archive()

```
void GiNaC::constant::archive (
            archive_node & n) const  [override], [virtual]
```

Save (serialize) the object into archive node.

Archive the object.

Losely speaking, this method turns an expression into a byte stream (which can be saved and restored later on, or sent via network, etc.)

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_string(), and name.

8.23.3.8 read_archive()

```
void GiNaC::constant::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented from GiNaC::basic.

References GiNaC::Catalan, GiNaC::Euler, GiNaC::archive_node::find_string(), name, and GiNaC::Pi.

8.23.3.9 derivative()

```
ex GiNaC::constant::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for a constant always returns 0.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, and GiNaC::basic::ex.

8.23.3.10 is_equal_same_type()

```
bool GiNaC::constant::is_equal_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if two objects of same type are equal.

Normally needs not be reimplemented as long as it wasn't overwritten by some parent class, since it just calls compare_same_type(). The reason why this function exists is that sometimes it is easier to determine equality than an order relation and then it can be overridden.

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic(), constant(), GINAC_ASSERT, GiNaC::is_exactly_a(), and serial.

8.23.3.11 calchash()

```
unsigned GiNaC::constant::calchash () const  [override], [protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References GiNaC::golden_ratio_hash(), GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, serial, and GiNaC::basic::setflag().

8.23.3.12 get_name()

```
std::string GiNaC::constant::get_name () const  [inline]
```

References name.

8.23.3.13 get_TeX_name()

```
std::string GiNaC::constant::get_TeX_name () const  [inline]
```

References TeX_name.

8.23.3.14 do_print()

```
void GiNaC::constant::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References name, and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.23.3.15 do_print_tree()

```
void GiNaC::constant::do_print_tree (
             const print_tree & c,
             unsigned level) const  [protected]
```

References GiNaC::basic::flags, GiNaC::basic::hashvalue, name, and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.23.3.16 do_print_latex()

```
void GiNaC::constant::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

References GiNaC::print_context::s, and TeX_name.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.23.3.17 do_print_python_repr()

```
void GiNaC::constant::do_print_python_repr (
             const print_python_repr & c,
             unsigned level) const  [protected]
```

References name, GiNaC::print_context::s, and TeX_name.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.23.4 Member Data Documentation

8.23.4.1 name

```
std::string GiNaC::constant::name  [private]
```

printname of this constant

Referenced by archive(), constant(), constant(), do_print(), do_print_python_repr(), do_print_tree(), get_name(), and read_archive().

8.23.4.2 TeX_name

```
std::string GiNaC::constant::TeX_name  [private]
```

LaTeX name.

Referenced by constant(), constant(), do_print_latex(), do_print_python_repr(), and get_TeX_name().

8.23.4.3 ef

```
evalffunctype GiNaC::constant::ef  [private]
```

Referenced by constant(), constant(), and evalf().

8.23.4.4 number

```
ex GiNaC::constant::number  [private]
```

numerical value this constant evalf()s to

Referenced by constant(), and evalf().

8.23.4.5 serial

```
unsigned GiNaC::constant::serial  [private]
```

unique serial number for comparison

Referenced by calchash(), constant(), constant(), and is_equal_same_type().

8.23.4.6 next_serial

```
unsigned GiNaC::constant::next_serial = 0  [static], [private]
```

Referenced by constant(), and constant().

8.23.4.7 domain

```
unsigned GiNaC::constant::domain  [private]
```

numerical value this constant evalf()s to

Referenced by conjugate(), constant(), constant(), imag_part(), info(), and real_part().

The documentation for this class was generated from the following files:

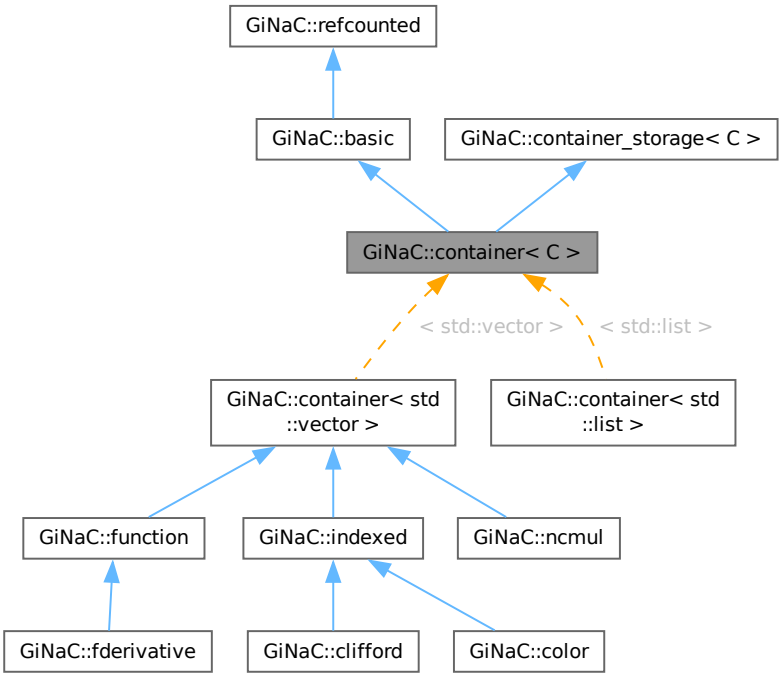
- constant.h
- constant.cpp

8.24 GiNaC::container< C > Class Template Reference

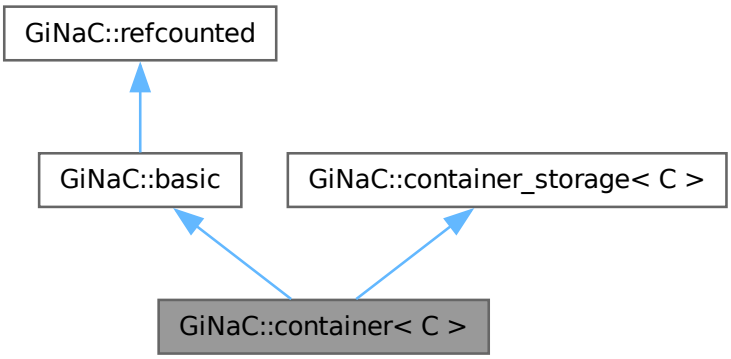
Wrapper template for making GiNaC classes out of STL containers.

```
#include <container.h>
```

Inheritance diagram for GiNaC::container< C >:



Collaboration diagram for GiNaC::container< C >:



Public Types

- typedef STLT::const_iterator const_iterator
- typedef STLT::const_reverse_iterator const_reverse_iterator

Public Member Functions

- container (STLT const &s)
- container (STLT &&v)
- container (exvector::const_iterator b, exvector::const_iterator e)
- container (std::initializer_list< ex > il)
- bool info (unsigned inf) const override

  Information about the object.
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- void read_archive (const archive_node &n, lst &sym_lst) override

  Load (deserialize) the object from an archive node.
- void archive (archive_node &n) const override

  Archive the object.
- container & prepend (const ex &b)

  Add element at front.
- container & append (const ex &b)

  Add element at back.

- container & remove_first ()

  Remove first element.
- container & remove_last ()

  Remove last element.
- container & remove_all ()

  Remove all elements.
- container & sort ()

  Sort elements.
- container & unique ()

  Remove adjacent duplicate elements.
- const_iterator begin () const
- const_iterator end () const
- const_reverse_iterator rbegin () const
- const_reverse_iterator rend () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const

- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Types

- typedef container_storage< C >::STLT STLT

Protected Types inherited from GiNaC::container_storage< C >

- typedef C< ex > STLT

Protected Member Functions

- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- virtual ex thiscontainer (const STLT &v) const

  Similar to duplicate(), but with a preset sequence.
- virtual ex thiscontainer (STLT &&v) const

  Similar to duplicate(), but with a preset sequence (which gets pilfered).
- virtual void printseq (const print_context &c, char openbracket, char delim, char closebracket, unsigned this↩
  _precedence, unsigned upper_precedence=0) const

  Print sequence of contained elements.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python (const print_python &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const
- STLT subschildren (const exmap &m, unsigned options=0) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Member Functions inherited from GiNaC::container_storage< C >

- container_storage ()
- container_storage (size_t n, const ex &e)
- container_storage (std::initializer_list< ex > il)
- template<class In>
  container_storage (In b, In e)
- void reserve (size_t)
- ~container_storage ()
- void reserve (size_t n)
- void reserve (std::vector< ex > &v, size_t n)

Static Protected Member Functions

- static unsigned get_default_flags ()
- static char get_open_delim ()
- static char get_close_delim ()

Static Protected Member Functions inherited from GiNaC::container_storage< C >

- static void reserve (STLT &, size_t)

Private Member Functions

- void sort_ (std::random_access_iterator_tag)
- void sort_ (std::input_iterator_tag)
- void unique_ ()
- void unique_ ()

  Specialization of container::unique_() for std::list.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

Protected Attributes inherited from GiNaC::container_storage< C >

- STLT seq

8.24.1 Detailed Description

template<template< class T, class=std::allocator< T > > class C>
class GiNaC::container< C >

Wrapper template for making GiNaC classes out of STL containers.

8.24.2 Member Typedef Documentation

8.24.2.1 STLT

```
template<template< class T, class=std::allocator< T > > class C>
typedef container_storage<C>::STLT GiNaC::container< C >::STLT  [protected]
```

8.24.2.2 const_iterator

```
template<template< class T, class=std::allocator< T > > class C>
typedef STLT::const_iterator GiNaC::container< C >::const_iterator
```

8.24.2.3 const_reverse_iterator

```
template<template< class T, class=std::allocator< T > > class C>
typedef STLT::const_reverse_iterator GiNaC::container< C >::const_reverse_iterator
```

8.24.3 Constructor & Destructor Documentation

8.24.3.1 container() [1/4]

```
template<template< class T, class=std::allocator< T > > class C>
GiNaC::container< C >::container (
            STLT const & s)  [inline]
```

Referenced by append(), is_equal_same_type(), prepend(), remove_all(), remove_first(), remove_last(), sort(), subs(), GiNaC::container< std::vector >::thiscontainer(), GiNaC::container< std::vector >::thiscontainer(), and unique().

8.24.3.2 container() [2/4]

```
template<template< class T, class=std::allocator< T > > class C>
GiNaC::container< C >::container (
            STLT && v)  [inline], [explicit]
```

8.24.3.3 container() [3/4]

```
template<template< class T, class=std::allocator< T > > class C>
GiNaC::container< C >::container (
            exvector::const_iterator b,
            exvector::const_iterator e)  [inline]
```

8.24.3.4 container() [4/4]

```
template<template< class T, class=std::allocator< T > > class C>
GiNaC::container< C >::container (
            std::initializer_list< ex > il)  [inline]
```

8.24.4 Member Function Documentation

8.24.4.1 get_default_flags()

```
template<template< class T, class=std::allocator< T > > class C>
unsigned GiNaC::container< C >::get_default_flags ()  [inline], [static], [protected]
```

8.24.4.2 get_open_delim()

```
template<template< class T, class=std::allocator< T > > class C>
char GiNaC::container< C >::get_open_delim ()  [inline], [static], [protected]
```

Referenced by do_print().

8.24.4.3 get_close_delim()

```
template<template< class T, class=std::allocator< T > > class C>
char GiNaC::container< C >::get_close_delim ()  [inline], [static], [protected]
```

Referenced by do_print().

8.24.4.4 info()

```
template<template< class T, class=std::allocator< T > > class C>
template bool GiNaC::container< C >::info (
            unsigned inf) const  [inline], [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::function, GiNaC::indexed, and GiNaC::ncmul.

8.24.4.5 precedence()

```
template<template< class T, class=std::allocator< T > > class C>
unsigned GiNaC::container< C >::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::function, GiNaC::indexed, and GiNaC::ncmul.

Referenced by do_print(), do_print_python(), and do_print_python_repr().

8.24.4.6 nops()

```
template<template< class T, class=std::allocator< T > > class C>
size_t GiNaC::container< C >::nops () const  [inline], [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

Referenced by GiNaC::ex::denom(), GiNaC::diag_matrix(), do_print_tree(), GiNaC::G2_eval(), GiNaC::G2_evalf(), GiNaC::G3_eval(), GiNaC::G3_evalf(), GiNaC::H_eval(), GiNaC::H_evalf(), GiNaC::is_discriminant_of_quadratic_number_field(), GiNaC::iterated_integral_evalf_impl(), GiNaC::Gt::kernel::kernel(), let_op(), GiNaC::Li_eval(), GiNaC::Li_evalf(), GiNaC::Gt::lst_evaluate(), GiNaC::lst_to_matrix(), GiNaC::add::normal(), GiNaC::basic::normal(), GiNaC::ex::normal(), GiNaC::mul::normal(), GiNaC::power::normal(), GiNaC::ex::numer(), GiNaC::ex::numer_denom(), op(), GiNaC::sqrfree(), and GiNaC::ex::subs().

8.24.4.7 op()

```
template<template< class T, class=std::allocator< T > > class C>
ex GiNaC::container< C >::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GINAC_ASSERT, nops(), and GiNaC::container_storage< C >::seq.

Referenced by GiNaC::clifford_moebius_map(), GiNaC::ex::denom(), GiNaC::mul::expand(), GiNaC::G2_eval(), GiNaC::G2_evalf(), GiNaC::G3_eval(), GiNaC::G3_evalf(), GiNaC::H_evalf(), GiNaC::is_discriminant_of_quadratic_number_field(), GiNaC::Gt::kernel::kernel(), GiNaC::kronecker_symbol(), GiNaC::Li_evalf(), GiNaC::Gt::lst_evaluate(), GiNaC::add::normal(), GiNaC::basic::normal(), GiNaC::ex::normal(), GiNaC::mul::normal(), GiNaC::power::normal(), GiNaC::ex::numer(), GiNaC::ex::numer_denom(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::sqrfree(), GiNaC::symm(), and GiNaC::symmetrize_cyclic().

8.24.4.8 let_op()

```
template<template< class T, class=std::allocator< T > > class C>
ex & GiNaC::container< C >::let_op (
            size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ensure_if_modifiable(), GiNaC::basic::ex, GINAC_ASSERT, nops(), and GiNaC::container_storage< C >::

Referenced by GiNaC::H_evalf().

8.24.4.9 subs()

```
template<template< class T, class=std::allocator< T > > class C>
ex GiNaC::container< C >::subs (
            const exmap & m,
            unsigned options = 0) const  [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References container(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::is_a(), GiNaC::basic::subs_one_level(), subschildren(), and thiscontainer().

8.24.4.10 read_archive()

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::read_archive (
            const archive_node & n,
            lst & syms)  [inline], [override], [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::fderivative, GiNaC::function, and GiNaC::indexed.

8.24.4.11 archive()

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::archive (
            archive_node & n) const  [inline], [override], [virtual]
```

Archive the object.

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::fderivative, GiNaC::function, and GiNaC::indexed.

8.24.4.12 conjugate()

```
template<template< class T, class=std::allocator< T > > class C>
ex GiNaC::container< C >::conjugate () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::function, and GiNaC::ncmul.

8.24.4.13 real_part()

```
template<template< class T, class=std::allocator< T > > class C>
ex GiNaC::container< C >::real_part () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::function, GiNaC::indexed, and GiNaC::ncmul.

8.24.4.14 imag_part()

```
template<template< class T, class=std::allocator< T > > class C>
ex GiNaC::container< C >::imag_part () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::function, GiNaC::indexed, and GiNaC::ncmul.

8.24.4.15 is_equal_same_type()

```
template<template< class T, class=std::allocator< T > > class C>
bool GiNaC::container< C >::is_equal_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if two objects of same type are equal.

Normally needs not be reimplemented as long as it wasn't overwritten by some parent class, since it just calls compare_same_type(). The reason why this function exists is that sometimes it is easier to determine equality than an order relation and then it can be overridden.

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::fderivative, and GiNaC::function.

References GiNaC::basic::basic(), container(), GINAC_ASSERT, GiNaC::is_a(), and GiNaC::container_storage< C >::seq.

8.24.4.16 thiscontainer() [1/2]

```
template<template< class T, class=std::allocator< T > > class C>
virtual ex GiNaC::container< C >::thiscontainer (
            const STLT & v) const  [inline], [protected], [virtual]
```

Similar to duplicate(), but with a preset sequence.

Must be overridden by derived classes.

Reimplemented in GiNaC::clifford, GiNaC::color, GiNaC::fderivative, GiNaC::function, GiNaC::indexed, and GiNaC::ncmul.

Referenced by subs().

8.24.4.17 thiscontainer() [2/2]

```
template<template< class T, class=std::allocator< T > > class C>
virtual ex GiNaC::container< C >::thiscontainer (
            STLT && v) const  [inline], [protected], [virtual]
```

Similar to duplicate(), but with a preset sequence (which gets pilfered).

Must be overridden by derived classes.

Reimplemented in GiNaC::clifford, GiNaC::color, GiNaC::fderivative, GiNaC::function, GiNaC::indexed, and GiNaC::ncmul.

8.24.4.18 printseq()

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::printseq (
            const print_context & c,
            char openbracket,
            char delim,
            char closebracket,
            unsigned this_precedence,
            unsigned upper_precedence = 0) const  [protected], [virtual]
```

Print sequence of contained elements.

References GiNaC::print_context::s, and GiNaC::container_storage< C >::seq.

Referenced by do_print(), do_print_python(), and do_print_python_repr().

8.24.4.19 sort_() [1/2]

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::sort_ (
            std::random_access_iterator_tag )  [inline], [private]
```

Referenced by sort().

8.24.4.20 sort_() [2/2]

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::sort_ (
            std::input_iterator_tag )  [inline], [private]
```

8.24.4.21 unique_() [1/2]

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::unique_ ()  [inline], [private]
```

Referenced by unique().

8.24.4.22 prepend()

```
template<template< class T, class=std::allocator< T > > class C>
container< C > & GiNaC::container< C >::prepend (
            const ex & b)
```

Add element at front.

References container(), GiNaC::basic::ensure_if_modifiable(), GiNaC::basic::ex, and GiNaC::container_storage< C >::seq.

8.24.4.23 append()

```
template<template< class T, class=std::allocator< T > > class C>
container< C > & GiNaC::container< C >::append (
            const ex & b)
```

Add element at back.

References container(), GiNaC::basic::ensure_if_modifiable(), GiNaC::basic::ex, and GiNaC::container_storage< C >::seq.

Referenced by GiNaC::Gt::archive(), GiNaC::clifford_to_lst(), GiNaC::Gt_detail::pathintegral_term::G_path(), GiNaC::H_evalf(), GiNaC::ifactor(), GiNaC::Gt_detail::TransformExpressionWithCache< Type >::impl(), GiNaC::Li_eval(), GiNaC::lsolve(), GiNaC::Gt::lst_evaluate(), GiNaC::symbol::read_archive(), GiNaC::replace_with_symbol(), GiNaC::S_eval(), GiNaC::sqrfree(), GiNaC::symm(), and GiNaC::symmetrize_cyclic().

8.24.4.24 remove_first()

```
template<template< class T, class=std::allocator< T > > class C>
container< C > & GiNaC::container< C >::remove_first ()
```

Remove first element.

References container(), GiNaC::basic::ensure_if_modifiable(), and GiNaC::container_storage< C >::seq.

Referenced by GiNaC::H_deriv(), GiNaC::sqrfree(), and GiNaC::symmetrize_cyclic().

8.24.4.25 remove_last()

```
template<template< class T, class=std::allocator< T > > class C>
container< C > & GiNaC::container< C >::remove_last ()
```

Remove last element.

References container(), GiNaC::basic::ensure_if_modifiable(), and GiNaC::container_storage< C >::seq.

8.24.4.26 remove_all()

```
template<template< class T, class=std::allocator< T > > class C>
container< C > & GiNaC::container< C >::remove_all ()
```

Remove all elements.

References container(), GiNaC::basic::ensure_if_modifiable(), and GiNaC::container_storage< C >::seq.

Referenced by GiNaC::clifford_to_lst().

8.24.4.27 sort()

```
template<template< class T, class=std::allocator< T > > class C>
container< C > & GiNaC::container< C >::sort ()
```

Sort elements.

References container(), GiNaC::basic::ensure_if_modifiable(), and sort_().

8.24.4.28 unique()

```
template<template< class T, class=std::allocator< T > > class C>
container< C > & GiNaC::container< C >::unique ()
```

Remove adjacent duplicate elements.

References container(), GiNaC::basic::ensure_if_modifiable(), and unique_().

8.24.4.29 begin()

```
template<template< class T, class=std::allocator< T > > class C>
const_iterator GiNaC::container< C >::begin () const  [inline]
```

Referenced by GiNaC::ex::antisymmetrize(), GiNaC::G3_eval(), GiNaC::G3_evalf(), GiNaC::H_deriv(), GiNaC::H_eval(), GiNaC::H_evalf(), GiNaC::H_print_latex(), GiNaC::kronecker_symbol(), GiNaC::Li_eval(), GiNaC::Li_evalf(), GiNaC::Li_print_latex(), GiNaC::ex::subs(), GiNaC::ex::symmetrize(), GiNaC::ex::symmetrize_cyclic(), GiNaC::zeta1_evalf(), GiNaC::zeta1_print_latex(), GiNaC::zeta2_evalf(), and GiNaC::zeta2_print_latex().

8.24.4.30 end()

```
template<template< class T, class=std::allocator< T > > class C>
const_iterator GiNaC::container< C >::end () const  [inline]
```

Referenced by GiNaC::ex::antisymmetrize(), do_print_tree(), GiNaC::G3_eval(), GiNaC::G3_evalf(), GiNaC::H_eval(), GiNaC::H_evalf(), GiNaC::H_print_latex(), GiNaC::kronecker_symbol(), GiNaC::Li_eval(), GiNaC::Li_evalf(), GiNaC::Li_print_latex(), GiNaC::ex::subs(), subschildren(), GiNaC::ex::symmetrize(), GiNaC::ex::symmetrize_cyclic(), GiNaC::zeta1_print_latex(), and GiNaC::zeta2_print_latex().

8.24.4.31 rbegin()

```
template<template< class T, class=std::allocator< T > > class C>
const_reverse_iterator GiNaC::container< C >::rbegin () const  [inline]
```

8.24.4.32 rend()

```
template<template< class T, class=std::allocator< T > > class C>
const_reverse_iterator GiNaC::container< C >::rend () const  [inline]
```

8.24.4.33 do_print()

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References get_close_delim(), get_open_delim(), precedence(), and printseq().

8.24.4.34 do_print_tree()

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::do_print_tree (
            const print_tree & c,
            unsigned level) const  [protected]
```

References GiNaC::print_tree::delta_indent, end(), GiNaC::basic::flags, GiNaC::basic::hashvalue, nops(), GiNaC::print_context::s, and GiNaC::container_storage< C >::seq.

8.24.4.35 do_print_python()

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::do_print_python (
            const print_python & c,
            unsigned level) const  [protected]
```

References precedence(), and printseq().

8.24.4.36 do_print_python_repr()

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container< C >::do_print_python_repr (
            const print_python_repr & c,
            unsigned level) const  [protected]
```

References precedence(), printseq(), and GiNaC::print_context::s.

8.24.4.37 subschildren()

```
template<template< class T, class=std::allocator< T > > class C>
container< C >::STLT GiNaC::container< C >::subschildren (
            const exmap & m,
            unsigned options = 0) const  [protected]
```

References GiNaC::are_ex_trivially_equal(), end(), GiNaC::basic::ex, GiNaC::container_storage< C >::reserve(), and GiNaC::container_storage< C >::seq.

Referenced by subs().

8.24.4.38 unique_() [2/2]

```
void GiNaC::container< std::list >::unique_ ()  [inline], [private]
```

Specialization of container::unique_() for std::list.

The documentation for this class was generated from the following files:

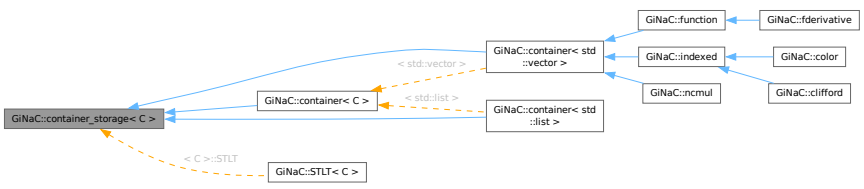
- container.h
- lst.h

8.25 GiNaC::container_storage< C > Class Template Reference

Helper template for encapsulating the reserve() mechanics of STL containers.

```
#include <container.h>
```

Inheritance diagram for GiNaC::container_storage< C >:



Protected Types

- typedef C< ex > STLT

Protected Member Functions

- container_storage ()
- container_storage (size_t n, const ex &e)
- container_storage (std::initializer_list< ex > il)
- template<class In>
  container_storage (In b, In e)
- void reserve (size_t)
- ~container_storage ()
- void reserve (size_t n)
- void reserve (std::vector< ex > &v, size_t n)

Static Protected Member Functions

- static void reserve (STLT &, size_t)

Protected Attributes

- STLT seq

8.25.1 Detailed Description

template<template< class T, class=std::allocator< T > > class C>
class GiNaC::container_storage< C >

Helper template for encapsulating the reserve() mechanics of STL containers.

8.25.2 Member Typedef Documentation

8.25.2.1 STLT

```
template<template< class T, class=std::allocator< T > > class C>
typedef C<ex> GiNaC::container_storage< C >::STLT  [protected]
```

8.25.3 Constructor & Destructor Documentation

8.25.3.1 container_storage() [1/4]

```
template<template< class T, class=std::allocator< T > > class C>
GiNaC::container_storage< C >::container_storage ()  [inline], [protected]
```

Referenced by GiNaC::container< std::vector >::container(), and GiNaC::container< std::vector >::container().

8.25.3.2 container_storage() [2/4]

```
template<template< class T, class=std::allocator< T > > class C>
GiNaC::container_storage< C >::container_storage (
            size_t n,
            const ex & e)  [inline], [protected]
```

8.25.3.3 container_storage() [3/4]

```
template<template< class T, class=std::allocator< T > > class C>
GiNaC::container_storage< C >::container_storage (
            std::initializer_list< ex > il)  [inline], [protected]
```

8.25.3.4 container_storage() [4/4]

```
template<template< class T, class=std::allocator< T > > class C>
template<class In>
GiNaC::container_storage< C >::container_storage (
            In b,
            In e)  [inline], [protected]
```

8.25.3.5 ~container_storage()

```
template<template< class T, class=std::allocator< T > > class C>
GiNaC::container_storage< C >::~container_storage ()  [inline], [protected]
```

8.25.4 Member Function Documentation

8.25.4.1 reserve() [1/4]

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container_storage< C >::reserve (
            size_t )  [inline], [protected]
```

Referenced by GiNaC::container< std::vector >::conjugate(), GiNaC::container< std::vector >::imag_part(), GiNaC::container< std::vector >::read_archive(), GiNaC::container< std::vector >::real_part(), and GiNaC::container< C >::subsch

8.25.4.2 reserve() [2/4]

```
template<template< class T, class=std::allocator< T > > class C>
void GiNaC::container_storage< C >::reserve (
            STLT & ,
            size_t )  [inline], [static], [protected]
```

8.25.4.3 reserve() [3/4]

```
void GiNaC::container_storage< std::vector >::reserve (
            size_t n)  [inline], [protected]
```

8.25.4.4 reserve() [4/4]

```
void GiNaC::container_storage< std::vector >::reserve (
            std::vector< ex > & v,
            size_t n)  [inline], [protected]
```

8.25.5 Member Data Documentation

8.25.5.1 seq

```
template<template< class T, class=std::allocator< T > > class C>
STLT GiNaC::container_storage< C >::seq  [protected]
```

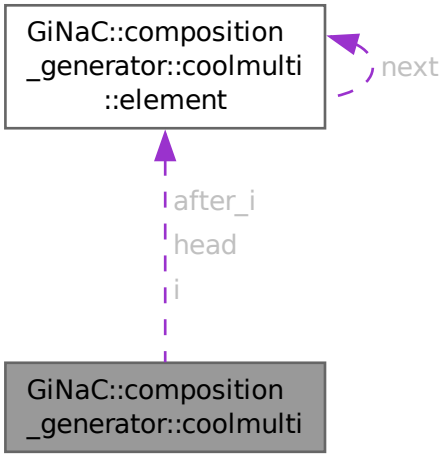
Referenced by GiNaC::indexed::all_index_values_are(), GiNaC::container< C >::append(), GiNaC::container< std::vector >::archive
GiNaC::container< std::vector >::begin(), GiNaC::ncmul::coeff(), GiNaC::container< std::vector >::conjugate(),
GiNaC::function::conjugate(), GiNaC::container< std::vector >::container(), GiNaC::container< std::vector >::container(),
GiNaC::ncmul::degree(), GiNaC::fderivative::derivative(), GiNaC::function::derivative(), GiNaC::ncmul::derivative(),
GiNaC::clifford::do_print_dflt(), GiNaC::clifford::do_print_latex(), GiNaC::clifford::do_print_tree(), GiNaC::container< C >::do_print_tre
GiNaC::fderivative::do_print_tree(), GiNaC::indexed::do_print_tree(), GiNaC::container< std::vector >::end(),
GiNaC::fderivative::eval(), GiNaC::function::eval(), GiNaC::indexed::eval(), GiNaC::ncmul::eval(), GiNaC::function::eval_ncmul(),
GiNaC::function::evalf(), GiNaC::ncmul::evalm(), GiNaC::function::expand(), GiNaC::indexed::expand(), GiNaC::ncmul::expand(),
GiNaC::ncmul::expandchildren(), GiNaC::function::expl_derivative(), GiNaC::indexed::get_dummy_indices(),
GiNaC::ncmul::get_factors(), GiNaC::indexed::get_free_indices(), GiNaC::indexed::get_indices(), GiNaC::indexed::has_dummy_index
GiNaC::function::imag_part(), GiNaC::indexed::indexed(), GiNaC::indexed::indexed(), GiNaC::function::info(),
GiNaC::indexed::info(), GiNaC::remember_table_entry::is_equal(), GiNaC::container< C >::is_equal_same_type(),
GiNaC::ncmul::ldegree(), GiNaC::container< C >::let_op(), GiNaC::container< std::vector >::nops(), GiNaC::container< C >::op(),
GiNaC::function::pderivative(), GiNaC::function::power(), GiNaC::container< C >::prepend(), GiNaC::function::print(),
GiNaC::indexed::print_indexed(), GiNaC::indexed::printindices(), GiNaC::container< C >::printseq(), GiNaC::container< std::vector >
GiNaC::container< std::vector >::read_archive(), GiNaC::function::read_archive(), GiNaC::indexed::read_archive(),
GiNaC::function::real_part(), GiNaC::container< C >::remove_all(), GiNaC::container< C >::remove_first(),
GiNaC::container< C >::remove_last(), GiNaC::container< std::vector >::rend(), GiNaC::indexed::reposition_dummy_indices,
GiNaC::function::return_type(), GiNaC::ncmul::return_type(), GiNaC::function::return_type_tinfo(), GiNaC::ncmul::return_type_tinfo(),
GiNaC::function::series(), GiNaC::indexed::simplify_indexed, GiNaC::container< std::vector >::sort_(), GiNaC::container< std::vecto
GiNaC::container< C >::subschildren(), GiNaC::container< std::vector >::unique_(), and GiNaC::indexed::validate().

The documentation for this class was generated from the following file:

- container.h

8.26 GiNaC::composition_generator::coolmulti Struct Reference

Collaboration diagram for GiNaC::composition_generator::coolmulti:



Classes

- struct element

Public Member Functions

- coolmulti (const std::vector< unsigned > &partition)
- ~coolmulti ()
- void next_permutation ()
- bool finished () const

Public Attributes

- element * head
- element * i
- element * after_i

8.26.1 Constructor & Destructor Documentation

8.26.1.1 coolmulti()

```
GiNaC::composition_generator::coolmulti::coolmulti (
            const std::vector< unsigned > & partition)  [inline], [explicit]
```

References after_i, GiNaC::composition_generator::coolmulti::element::element(), head, i, and GiNaC::composition_generator::coolm

8.26.1.2 ~coolmulti()

```
GiNaC::composition_generator::coolmulti::~coolmulti ()  [inline]
```

References head.

8.26.2 Member Function Documentation

8.26.2.1 next_permutation()

```
void GiNaC::composition_generator::coolmulti::next_permutation ()  [inline]
```

References after_i, head, i, GiNaC::composition_generator::coolmulti::element::next, and GiNaC::composition_generator::coolmulti::e

Referenced by GiNaC::composition_generator::next().

8.26.2.2 finished()

```
bool GiNaC::composition_generator::coolmulti::finished () const  [inline]
```

References after_i, head, GiNaC::composition_generator::coolmulti::element::next, and GiNaC::composition_generator::coolmulti::ele

Referenced by GiNaC::composition_generator::next().

8.26.3 Member Data Documentation

8.26.3.1 head

```
element* GiNaC::composition_generator::coolmulti::head
```

Referenced by coolmulti(), finished(), GiNaC::composition_generator::get(), next_permutation(), and ~coolmulti().

8.26.3.2 i

```
element * GiNaC::composition_generator::coolmulti::i
```

Referenced by coolmulti(), and next_permutation().

8.26.3.3 after_i

```
element * GiNaC::composition_generator::coolmulti::after_i
```

Referenced by coolmulti(), finished(), and next_permutation().

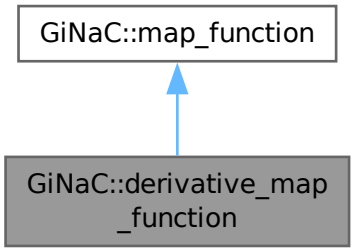
The documentation for this struct was generated from the following file:

- utils.h

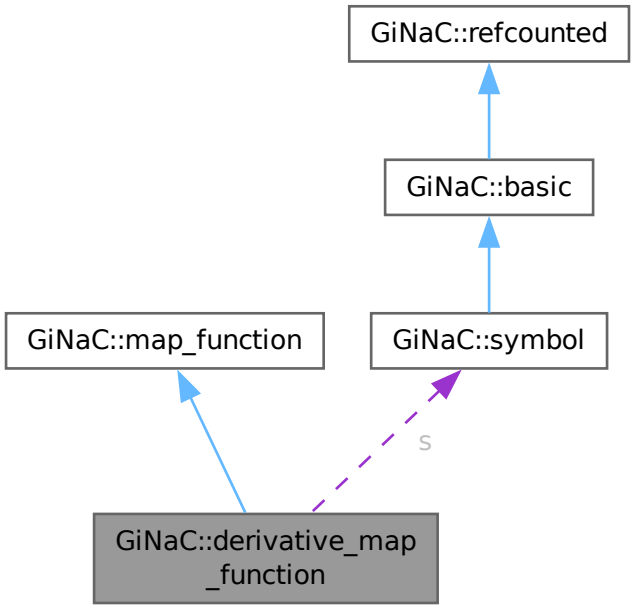
8.27 GiNaC::derivative_map_function Struct Reference

Function object to be applied by basic::derivative().

Inheritance diagram for GiNaC::derivative_map_function:



Collaboration diagram for GiNaC::derivative_map_function:



Public Member Functions

- derivative_map_function (const symbol &sym)
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Public Attributes

- const symbol & s

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.27.1 Detailed Description

Function object to be applied by basic::derivative().

8.27.2 Constructor & Destructor Documentation

8.27.2.1 derivative_map_function()

```
GiNaC::derivative_map_function::derivative_map_function (
            const symbol & sym)  [inline]
```

References s.

8.27.3 Member Function Documentation

8.27.3.1 operator()()

```
ex GiNaC::derivative_map_function::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References GiNaC::diff(), and s.

8.27.4 Member Data Documentation

8.27.4.1 s

```
const symbol& GiNaC::derivative_map_function::s
```

Referenced by derivative_map_function(), and operator()().

The documentation for this struct was generated from the following file:

- basic.cpp

8.28 GiNaC::determinant_algo Class Reference

Switch to control algorithm for determinant computation.

`#include <flags.h>`

Public Types

- enum {
  automatic , gauss , divfree , laplace ,
  bareiss }

8.28.1 Detailed Description

Switch to control algorithm for determinant computation.

8.28.2 Member Enumeration Documentation

8.28.2.1 anonymous enum

`anonymous enum`

Enumerator

automatic	Let the system choose. A heuristics is applied for automatic determination of a suitable algorithm.
gauss	Gauss elimination. If $m_{i,j}^{(0)}$ are the entries of the original matrix, then the matrix is transformed into triangular form by applying the rules $$m_{i,j}^{(k+1)} = m_{i,j}^{(k)} - m_{i,k}^{(k)} m_{k,j}^{(k)} / m_{k,k}^{(k)}$$ The determinant is then just the product of diagonal elements. Choose this algorithm only for purely numerical matrices.
divfree	Division-free elimination. This is a modification of Gauss elimination where the division by the pivot element is not carried out. If $m_{i,j}^{(0)}$ are the entries of the original matrix, then the matrix is transformed into triangular form by applying the rules $$m_{i,j}^{(k+1)} = m_{i,j}^{(k)} m_{k,k}^{(k)} - m_{i,k}^{(k)} m_{k,j}^{(k)}$$ The determinant can later be computed by inspecting the diagonal elements only. This algorithm is only there for the purpose of cross-checks. It is never fast.
laplace	Laplace elimination. This is plain recursive elimination along minors although multiple minors are avoided by the algorithm. Although the algorithm is exponential in complexity it is frequently the fastest one when the matrix is populated by complicated symbolic expressions.

bareiss	Bareiss fraction-free elimination. This is a modification of Gauss elimination where the division by the pivot element is *delayed* until it can be carried out without computing GCDs. If $m_{i,j}^{(0)}$ are the entries of the original matrix, then the matrix is transformed into triangular form by applying the rules $$m_{i,j}^{(k+1)} = (m_{i,j}^{(k)} m_{k,k}^{(k)} - m_{i,k}^{(k)} m_{k,j}^{(k)}) / m_{k-1,k-1}^{(k-1)}$$ (We have set $m_{-1,-1}^{(-1)} = 1$ in order to avoid a case distinction in above formula.) It can be shown that nothing more than polynomial long division is needed for carrying out the division. The determinant can then be read of from the lower right entry. This algorithm is rarely fast for computing determinants.

The documentation for this class was generated from the following file:

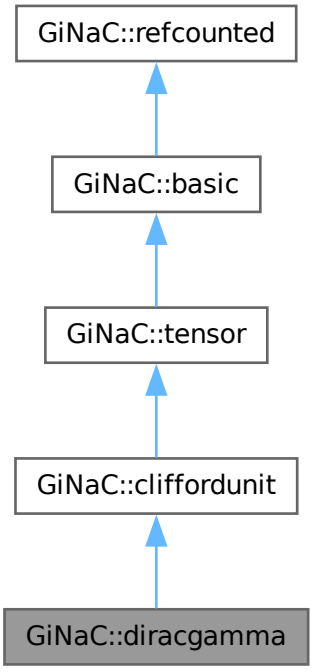
- flags.h

8.29 GiNaC::diracgamma Class Reference

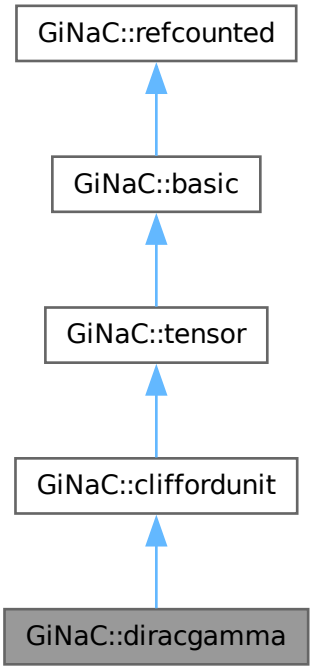
This class represents the Dirac gamma Lorentz vector.

`#include <clifford.h>`

Inheritance diagram for GiNaC::diracgamma:



Collaboration diagram for GiNaC::diracgamma:



Public Member Functions

- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of a gamma matrix with something else.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

Perform automatic non-interruptive term rewriting rules.

- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

Default implementation of ex::series().

- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::cliffordunit

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.29.1 Detailed Description

This class represents the Dirac gamma Lorentz vector.

8.29.2 Member Function Documentation

8.29.2.1 contract_with()

```
bool GiNaC::diracgamma::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [override], [virtual]
```

Contraction of a gamma matrix with something else.

Reimplemented from GiNaC::cliffordunit.

References GiNaC::_ex1, GiNaC::_ex2, GiNaC::_ex_2, GiNaC::base_and_index(), GiNaC::dirac_ONE(), GiNaC::dirac_slash(), GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::lorentz_g(), GiNaC::minimal_dim(), and GiNaC::ncmul::ncmul().

8.29.2.2 do_print()

```
void GiNaC::diracgamma::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

8.29.2.3 do_print_latex()

```
void GiNaC::diracgamma::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

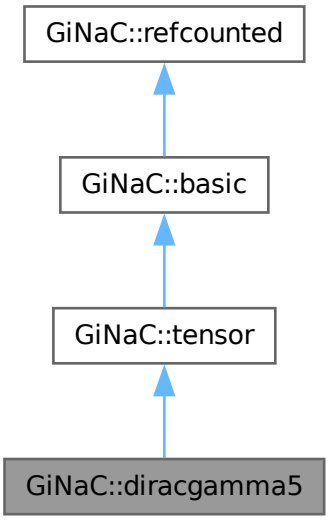
- clifford.h
- clifford.cpp

8.30 GiNaC::diracgamma5 Class Reference

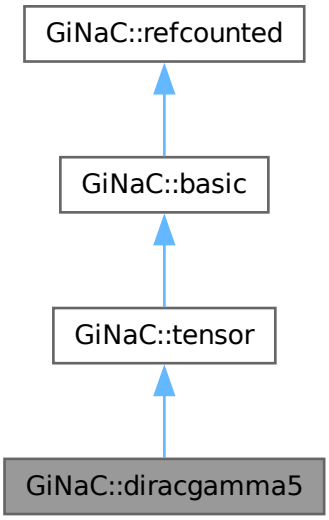
This class represents the Dirac gamma5 object which anticommutates with all other gammas.

`#include <clifford.h>`

Inheritance diagram for GiNaC::diracgamma5:



Collaboration diagram for GiNaC::diracgamma5:



Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Member Functions

- ex conjugate () const override

Additional Inherited Members

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

Return degree of highest power in object s.

- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual return_type_t return_type_tinfo () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.

- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.30.1 Detailed Description

This class represents the Dirac gamma5 object which anticommutates with all other gammas.

8.30.2 Member Function Documentation

8.30.2.1 conjugate()

```
ex GiNaC::diracgamma5::conjugate () const  [override], [private], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::_ex_1.

8.30.2.2 do_print()

```
void GiNaC::diracgamma5::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

8.30.2.3 do_print_latex()

```
void GiNaC::diracgamma5::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

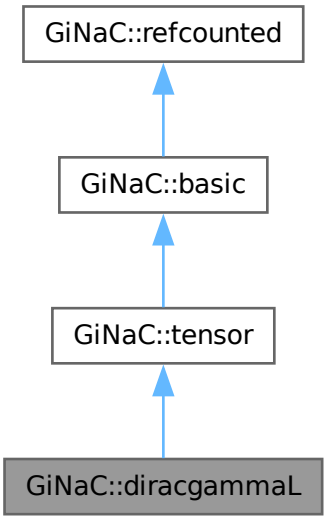
- clifford.h
- clifford.cpp

8.31 GiNaC::diracgammaL Class Reference

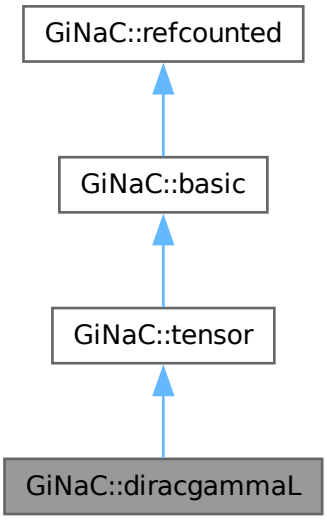
This class represents the Dirac gammaL object which behaves like 1/2 (1-gamma5).

`#include <clifford.h>`

Inheritance diagram for GiNaC::diracgammaL:



Collaboration diagram for GiNaC::diracgammaL:



Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Member Functions

- ex conjugate () const override

Additional Inherited Members

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*

- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.

- virtual basic * duplicate () const

  Create a clone of this object on the heap.

- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.

- virtual ex evalf () const

  Evaluate object numerically.

- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.

- virtual ex eval_integ () const

  Evaluate integrals, if result is known.

- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.

- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.

- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.

- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).

- virtual bool info (unsigned inf) const

  Information about the object.

- virtual size_t nops () const

  Number of operands/members.

- virtual ex op (size_t i) const

  Return operand/member at position i.

- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.

- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.

- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.

- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.

- virtual int degree (const ex &s) const

Return degree of highest power in object s.

- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual return_type_t return_type_tinfo () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.

- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.31.1 Detailed Description

This class represents the Dirac gammaL object which behaves like 1/2 (1-gamma5).

8.31.2 Member Function Documentation

8.31.2.1 conjugate()

```
ex GiNaC::diracgammaL::conjugate () const  [override], [private], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate().

8.31.2.2 do_print()

```
void GiNaC::diracgammaL::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

8.31.2.3 do_print_latex()

```
void GiNaC::diracgammaL::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

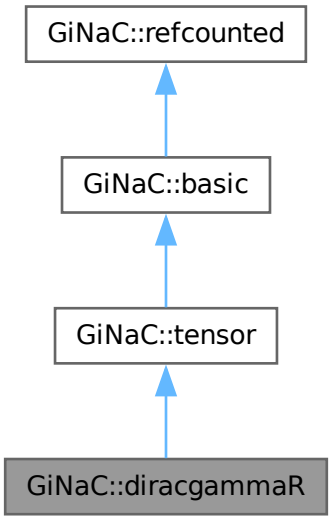
- clifford.h
- clifford.cpp

8.32 GiNaC::diracgammaR Class Reference

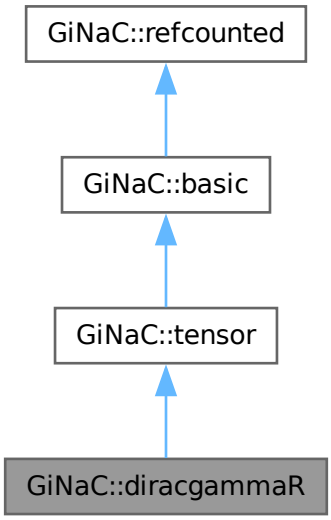
This class represents the Dirac gammaL object which behaves like 1/2 (1+gamma5).

`#include <clifford.h>`

Inheritance diagram for GiNaC::diracgammaR:



Collaboration diagram for GiNaC::diracgammaR:



Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Member Functions

- ex conjugate () const override

Additional Inherited Members

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

Return degree of highest power in object s.

- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual return_type_t return_type_tinfo () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.

- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.32.1 Detailed Description

This class represents the Dirac gammaL object which behaves like 1/2 (1+gamma5).

8.32.2 Member Function Documentation

8.32.2.1 conjugate()

```
ex GiNaC::diracgammaR::conjugate () const  [override], [private], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate().

8.32.2.2 do_print()

```
void GiNaC::diracgammaR::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

8.32.2.3 do_print_latex()

```
void GiNaC::diracgammaR::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

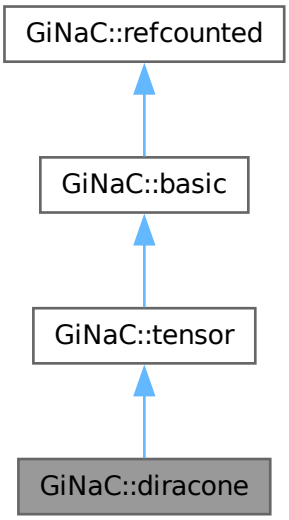
- clifford.h
- clifford.cpp

8.33 GiNaC::diracone Class Reference

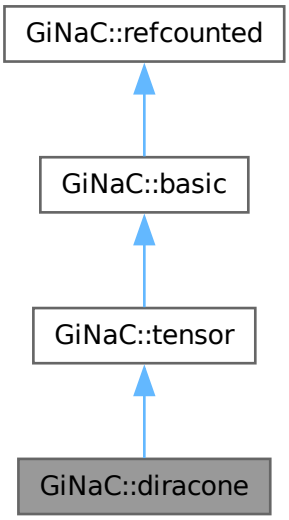
This class represents the Clifford algebra unity element.

`#include <clifford.h>`

Inheritance diagram for GiNaC::diracone:



Collaboration diagram for GiNaC::diracone:



Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

Return degree of highest power in object s.

- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

Stop further evaluation.

- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.33.1 Detailed Description

This class represents the Clifford algebra unity element.

8.33.2 Member Function Documentation

8.33.2.1 do_print()

```
void GiNaC::diracone::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

8.33.2.2 do_print_latex()

```
void GiNaC::diracone::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

The documentation for this class was generated from the following file:

- clifford.h

8.34 GiNaC::do_taylor Class Reference

Exception class thrown by classes which provide their own series expansion to signal that ordinary Taylor expansion is safe.

`#include <function.h>`

8.34.1 Detailed Description

Exception class thrown by classes which provide their own series expansion to signal that ordinary Taylor expansion is safe.

The documentation for this class was generated from the following file:

- function.h

8.35 GiNaC::domain Class Reference

Domain of an object.

`#include <flags.h>`

Public Types

- enum { complex , real , positive }

8.35.1 Detailed Description

Domain of an object.

8.35.2 Member Enumeration Documentation

8.35.2.1 anonymous enum

`anonymous enum`

Enumerator

complex	
real	
positive	

The documentation for this class was generated from the following file:

- flags.h

8.36 GiNaC::dunno Class Reference

Exception class thrown by functions to signal unimplemented functionality so the expression may just be .hold().

`#include <utils.h>`

8.36.1 Detailed Description

Exception class thrown by functions to signal unimplemented functionality so the expression may just be .hold().

The documentation for this class was generated from the following file:

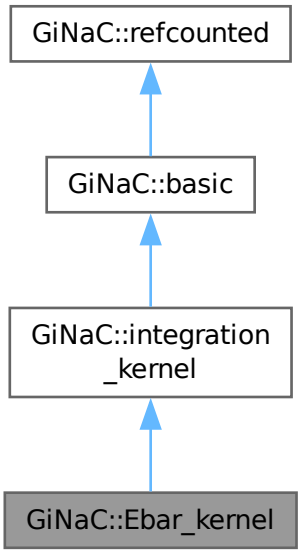
- utils.h

8.37 GiNaC::Ebar_kernel Class Reference

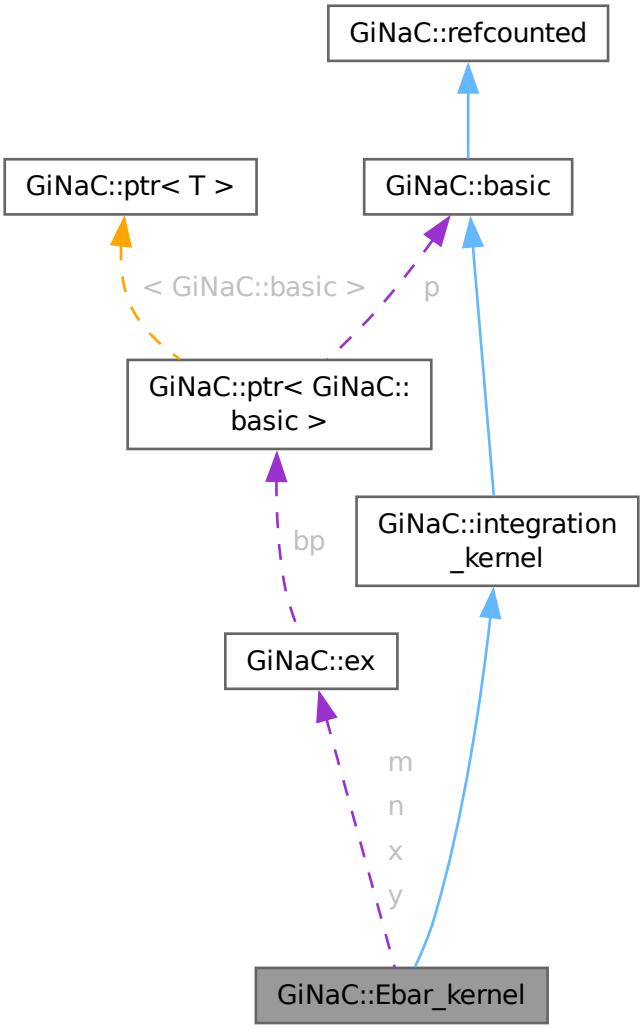
The Ebar-kernel.

`#include <integration_kernel.h>`

Inheritance diagram for GiNaC::Ebar_kernel:



Collaboration diagram for GiNaC::Ebar_kernel:



Public Member Functions

- Ebar_kernel (const ex &n, const ex &m, const ex &x, const ex &y)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- bool is_numeric (void) const override

  This routine returns true, if the integration kernel can be evaluated numerically.
- ex get_numerical_value (const ex &qbar, int N_trunc=0) const override

  Returns the value of Ebar_{n,m}(x,y,qbar).

Public Member Functions inherited from GiNaC::integration_kernel

- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- virtual ex Laurent_series (const ex &x, int order) const

  Returns the Laurent series, starting possibly with the pole term.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

Load (deserialize) the object from an archive node.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- cln::cl_N series_coeff_impl (int i) const override

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual bool uses_Laurent_series () const

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex n
- ex m
- ex x
- ex y

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.37.1 Detailed Description

The Ebar-kernel.

This class represents the differential one-form

$$\omega^{\overline{\mathrm{E}}}_{n;m}(x;y) = \overline{\mathrm{E}}_{n;m}(x;y;\bar{q})\frac{d\bar{q}}{\bar{q}}$$

8.37.2 Constructor & Destructor Documentation

8.37.2.1 Ebar_kernel()

```
GiNaC::Ebar_kernel::Ebar_kernel (
             const ex & n,
             const ex & m,
             const ex & x,
             const ex & y)
```

References Ebar_kernel(), m, n, x, and y.

Referenced by Ebar_kernel(), GiNaC::Kronecker_dtau_kernel::get_numerical_value(), and GiNaC::Kronecker_dz_kernel::series_coef

8.37.3 Member Function Documentation

8.37.3.1 nops()

```
size_t GiNaC::Ebar_kernel::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.37.3.2 op()

```
ex GiNaC::Ebar_kernel::op (
             size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References m, n, x, and y.

8.37.3.3 let_op()

```
ex & GiNaC::Ebar_kernel::let_op (
             size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ensure_if_modifiable(), m, n, x, and y.

8.37.3.4 is_numeric()

```
bool GiNaC::Ebar_kernel::is_numeric (
            void ) const  [override], [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented from GiNaC::integration_kernel.

References GiNaC::ex::evalf(), GiNaC::ex::info(), m, n, GiNaC::info_flags::nonnegint, GiNaC::info_flags::numeric, x, and y.

8.37.3.5 get_numerical_value()

```
ex GiNaC::Ebar_kernel::get_numerical_value (
            const ex & qbar,
            int N_trunc = 0) const  [override], [virtual]
```

Returns the value of Ebar_{n,m}(x,y,qbar).

Reimplemented from GiNaC::integration_kernel.

References GiNaC::integration_kernel::get_numerical_value_impl().

Referenced by GiNaC::Kronecker_dz_kernel::series_coeff_impl().

8.37.3.6 series_coeff_impl()

```
cln::cl_N GiNaC::Ebar_kernel::series_coeff_impl (
            int i) const  [override], [protected], [virtual]
```

For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.

The i-th coefficient corresponds to the power λ^{i-1}.

Reimplemented from GiNaC::integration_kernel.

References GiNaC::ex::evalf(), GiNaC::ex_to(), m, n, x, and y.

8.37.3.7 do_print()

```
void GiNaC::Ebar_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References m, n, GiNaC::ex::print(), GiNaC::print_context::s, x, and y.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.37.4 Member Data Documentation

8.37.4.1 n

```
ex GiNaC::Ebar_kernel::n  [protected]
```

Referenced by do_print(), Ebar_kernel(), is_numeric(), let_op(), op(), and series_coeff_impl().

8.37.4.2 m

```
ex GiNaC::Ebar_kernel::m  [protected]
```

Referenced by do_print(), Ebar_kernel(), is_numeric(), let_op(), op(), and series_coeff_impl().

8.37.4.3 x

```
ex GiNaC::Ebar_kernel::x  [protected]
```

Referenced by do_print(), Ebar_kernel(), is_numeric(), let_op(), op(), and series_coeff_impl().

8.37.4.4 y

```
ex GiNaC::Ebar_kernel::y  [protected]
```

Referenced by do_print(), Ebar_kernel(), is_numeric(), let_op(), op(), and series_coeff_impl().

The documentation for this class was generated from the following files:

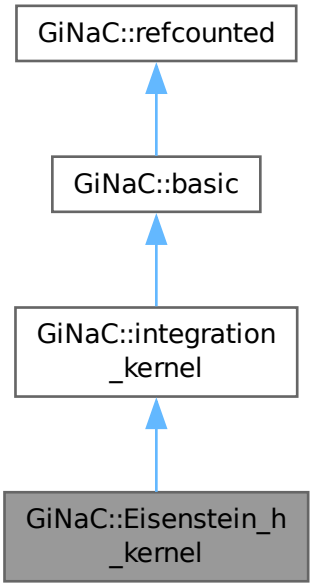
- integration_kernel.h
- integration_kernel.cpp

8.38 GiNaC::Eisenstein_h_kernel Class Reference

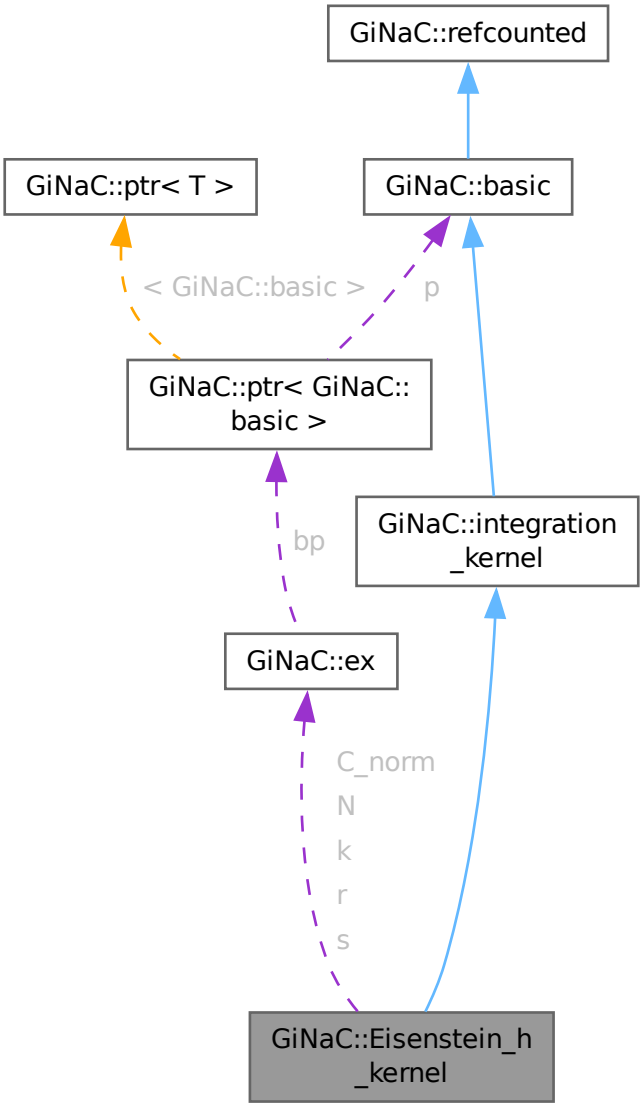
The kernel corresponding to the Eisenstein series $h_{k,N,r,s}(\tau)$.

```
#include <integration_kernel.h>
```

Inheritance diagram for GiNaC::Eisenstein_h_kernel:



Collaboration diagram for GiNaC::Eisenstein_h_kernel:



Public Member Functions

- Eisenstein_h_kernel (const ex &k, const ex &N, const ex &r, const ex &s, const ex &C_norm=numeric(1))
- ex series (const relational &r, int order, unsigned options=0) const override

  The series method for this class returns the qbar-expansion of the modular form, without an additional factor of C_norm/qbar.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

Return modifiable operand/member at position i.

- bool is_numeric (void) const override

  This routine returns true, if the integration kernel can be evaluated numerically.
- ex Laurent_series (const ex &x, int order) const override

  Returns the Laurent series, starting possibly with the pole term.
- ex get_numerical_value (const ex &qbar, int N_trunc=0) const override

  Returns the value of the modular form.
- ex coefficient_a0 (const numeric &k, const numeric &r, const numeric &s, const numeric &N) const

  The constant coefficient in the Fourier expansion.
- ex coefficient_an (const numeric &n, const numeric &k, const numeric &r, const numeric &s, const numeric &N) const

  The higher coefficients in the Fourier expansion.
- ex q_expansion_modular_form (const ex &q, int order) const

Public Member Functions inherited from GiNaC::integration_kernel

- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

Output to stream.

- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

Try to contract two indexed expressions that appear in the same product.

- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- bool uses_Laurent_series () const override

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual cln::cl_N series_coeff_impl (int i) const

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex k
- ex N
- ex r
- ex s
- ex C_norm

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.38.1 Detailed Description

The kernel corresponding to the Eisenstein series $h_{k,N,r,s}(\tau)$.

This class represents the differential one-form

$$\omega_{k,N,r,s}^{\mathrm{Eisenstein,h}} = C_k h_{k,N,r,s}(\tau)\frac{d\bar{q}_N}{\bar{q}_N}$$

8.38.2 Constructor & Destructor Documentation

8.38.2.1 Eisenstein_h_kernel()

```
GiNaC::Eisenstein_h_kernel::Eisenstein_h_kernel (
            const ex & k,
            const ex & N,
            const ex & r,
            const ex & s,
            const ex & C_norm = numeric(1))
```

References C_norm, Eisenstein_h_kernel(), k, N, r, and s.

Referenced by Eisenstein_h_kernel().

8.38.3 Member Function Documentation

8.38.3.1 series()

```
ex GiNaC::Eisenstein_h_kernel::series (
            const relational & r,
            int order,
            unsigned options = 0) const  [override], [virtual]
```

The series method for this class returns the qbar-expansion of the modular form, without an additional factor of C_norm/qbar.

This allows for easy use in the class modular_form_kernel.

Reimplemented from GiNaC::basic.

References GiNaC::relational::lhs(), q_expansion_modular_form(), r, GiNaC::relational::rhs(), and GiNaC::ex::series().

8.38.3.2 nops()

```
size_t GiNaC::Eisenstein_h_kernel::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.38.3.3 op()

```
ex GiNaC::Eisenstein_h_kernel::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References C_norm, k, N, r, and s.

8.38.3.4 let_op()

```
ex & GiNaC::Eisenstein_h_kernel::let_op (
             size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References C_norm, GiNaC::basic::ensure_if_modifiable(), k, N, r, and s.

8.38.3.5 is_numeric()

```
bool GiNaC::Eisenstein_h_kernel::is_numeric (
             void ) const  [override], [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented from GiNaC::integration_kernel.

References C_norm, GiNaC::ex::evalf(), GiNaC::ex::info(), GiNaC::info_flags::integer, k, N, GiNaC::info_flags::nonnegint, GiNaC::info_flags::numeric, GiNaC::info_flags::posint, r, and s.

8.38.3.6 Laurent_series()

```
ex GiNaC::Eisenstein_h_kernel::Laurent_series (
             const ex & x,
             int order) const  [override], [virtual]
```

Returns the Laurent series, starting possibly with the pole term.

Neglected terms are of order $O(x^order)$.

Reimplemented from GiNaC::integration_kernel.

References C_norm, q_expansion_modular_form(), and GiNaC::ex::series().

8.38.3.7 get_numerical_value()

```
ex GiNaC::Eisenstein_h_kernel::get_numerical_value (
             const ex & qbar,
             int N_trunc = 0) const  [override], [virtual]
```

Returns the value of the modular form.

Reimplemented from GiNaC::integration_kernel.

References C_norm, GiNaC::integration_kernel::get_numerical_value_impl(), and GiNaC::numeric::numeric().

8.38.3.8 uses_Laurent_series()

```
bool GiNaC::Eisenstein_h_kernel::uses_Laurent_series () const  [override], [protected], [virtual]
```

Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).

Returns false if this is not the case (and the class has an implementation of series_coeff_impl).

Reimplemented from GiNaC::integration_kernel.

8.38.3.9 coefficient_a0()

```
ex GiNaC::Eisenstein_h_kernel::coefficient_a0 (
             const numeric & k,
             const numeric & r,
             const numeric & s,
             const numeric & N) const
```

The constant coefficient in the Fourier expansion.

References GiNaC::Bernoulli_polynomial(), GiNaC::cos(), GiNaC::I, GiNaC::irem(), k, GiNaC::mod(), N, GiNaC::numeric::numeric(), GiNaC::Pi, r, s, and GiNaC::sin().

Referenced by q_expansion_modular_form().

8.38.3.10 coefficient_an()

```
ex GiNaC::Eisenstein_h_kernel::coefficient_an (
             const numeric & n,
             const numeric & k,
             const numeric & r,
             const numeric & s,
             const numeric & N) const
```

The higher coefficients in the Fourier expansion.

References GiNaC::exp(), GiNaC::I, GiNaC::irem(), k, GiNaC::mod(), N, GiNaC::numeric::numeric(), GiNaC::Pi, GiNaC::pow(), r, and s.

Referenced by q_expansion_modular_form().

8.38.3.11 q_expansion_modular_form()

```
ex GiNaC::Eisenstein_h_kernel::q_expansion_modular_form (
             const ex & q,
             int order) const
```

References coefficient_a0(), coefficient_an(), GiNaC::ex_to(), k, N, GiNaC::numeric::numeric(), GiNaC::pow(), r, s, and GiNaC::ex::series().

Referenced by Laurent_series(), and series().

8.38.3.12 do_print()

```
void GiNaC::Eisenstein_h_kernel::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References C_norm, k, N, GiNaC::ex::print(), r, s, and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.38.4 Member Data Documentation

8.38.4.1 k

```
ex GiNaC::Eisenstein_h_kernel::k  [protected]
```

Referenced by coefficient_a0(), coefficient_an(), do_print(), Eisenstein_h_kernel(), is_numeric(), let_op(), op(), and q_expansion_modular_form().

8.38.4.2 N

```
ex GiNaC::Eisenstein_h_kernel::N  [protected]
```

Referenced by coefficient_a0(), coefficient_an(), do_print(), Eisenstein_h_kernel(), is_numeric(), let_op(), op(), and q_expansion_modular_form().

8.38.4.3 r

```
ex GiNaC::Eisenstein_h_kernel::r  [protected]
```

Referenced by coefficient_a0(), coefficient_an(), do_print(), Eisenstein_h_kernel(), is_numeric(), let_op(), op(), q_expansion_modular_form(), and series().

8.38.4.4 s

```
ex GiNaC::Eisenstein_h_kernel::s  [protected]
```

Referenced by coefficient_a0(), coefficient_an(), do_print(), Eisenstein_h_kernel(), is_numeric(), let_op(), op(), and q_expansion_modular_form().

8.38.4.5 C_norm

```
ex GiNaC::Eisenstein_h_kernel::C_norm  [protected]
```

Referenced by do_print(), Eisenstein_h_kernel(), get_numerical_value(), is_numeric(), Laurent_series(), let_op(), and op().

The documentation for this class was generated from the following files:

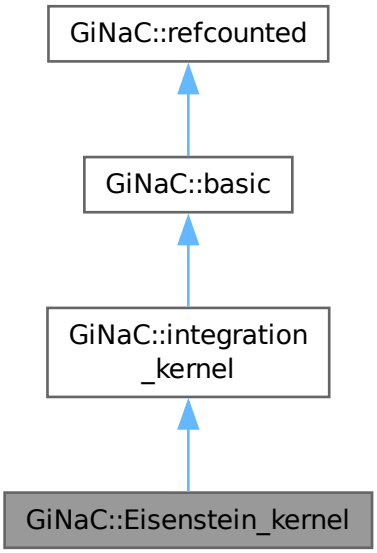
- integration_kernel.h
- integration_kernel.cpp

8.39 GiNaC::Eisenstein_kernel Class Reference

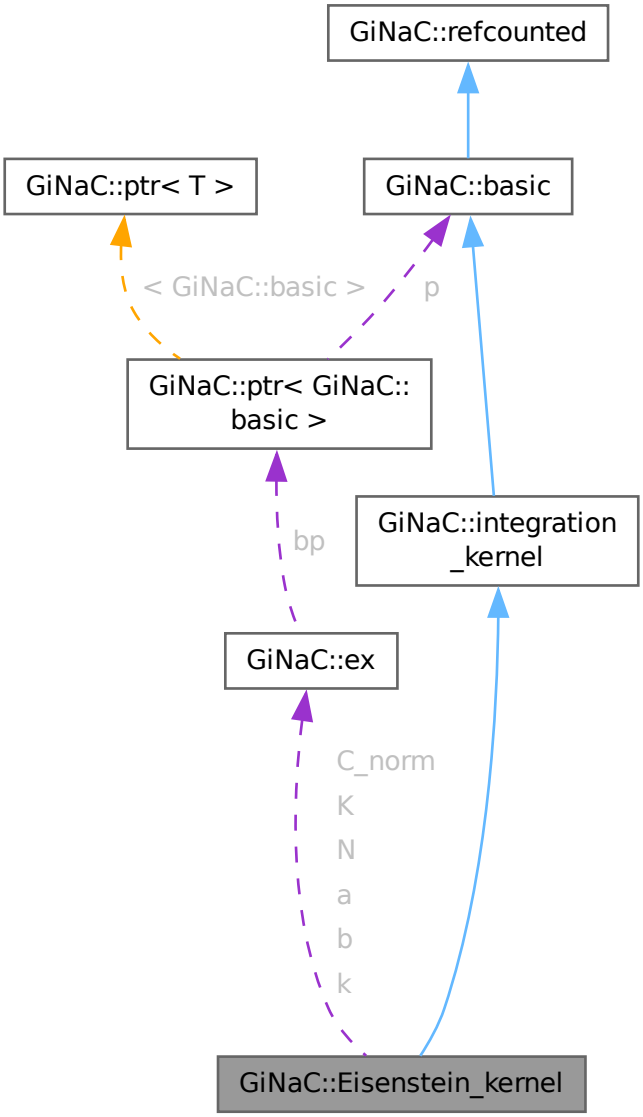
The kernel corresponding to the Eisenstein series $E_{k,N,a,b,K}(\tau)$.

`#include <integration_kernel.h>`

Inheritance diagram for GiNaC::Eisenstein_kernel:



Collaboration diagram for GiNaC::Eisenstein_kernel:



Public Member Functions

- Eisenstein_kernel (const ex &k, const ex &N, const ex &a, const ex &b, const ex &K, const ex &C_↩ norm=numeric(1))
- ex series (const relational &r, int order, unsigned options=0) const override

  The series method for this class returns the qbar-expansion of the modular form, without an additional factor of C_norm/qbar.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.

- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- bool is_numeric (void) const override

  This routine returns true, if the integration kernel can be evaluated numerically.
- ex Laurent_series (const ex &x, int order) const override

  Returns the Laurent series, starting possibly with the pole term.
- ex get_numerical_value (const ex &qbar, int N_trunc=0) const override

  Returns the value of the modular form.
- ex q_expansion_modular_form (const ex &q, int order) const

Public Member Functions inherited from GiNaC::integration_kernel

- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

Little wrapper around printtree to be called within a debugger.

- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const

- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- bool uses_Laurent_series () const override

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual cln::cl_N series_coeff_impl (int i) const

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex k
- ex N
- ex a
- ex b
- ex K
- ex C_norm

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.39.1 Detailed Description

The kernel corresponding to the Eisenstein series $E_{k,N,a,b,K}(\tau)$.

This class represents the differential one-form

$$\omega_{k,N,a,b,K}^{\text{Eisenstein}} = C_k E_{k,N,a,b,K}(\tau) \frac{d\bar{q}_N}{\bar{q}_N}$$

The integers a and b are either one or the discriminant of a quadratic number field. This class represents Eisenstein series, which can be defined by primitive Dirichlet characters from the Kronecker symbol. This implies that the characters take the values -1,0,1, i.e. no higher roots of unity occur. The

$$\bar{q}$$

-expansion has then rational coefficients.

Ref.: W. Stein, Modular Forms: A Computational Approach, Chapter 5

8.39.2 Constructor & Destructor Documentation

8.39.2.1 Eisenstein_kernel()

```
GiNaC::Eisenstein_kernel::Eisenstein_kernel (
             const ex & k,
             const ex & N,
             const ex & a,
             const ex & b,
             const ex & K,
             const ex & C_norm = numeric(1))
```

References a, b, C_norm, Eisenstein_kernel(), K, k, and N.

Referenced by Eisenstein_kernel().

8.39.3 Member Function Documentation

8.39.3.1 series()

```
ex GiNaC::Eisenstein_kernel::series (
             const relational & r,
             int order,
             unsigned options = 0) const  [override], [virtual]
```

The series method for this class returns the qbar-expansion of the modular form, without an additional factor of C_norm/qbar.

This allows for easy use in the class modular_form_kernel.

Reimplemented from GiNaC::basic.

References GiNaC::relational::lhs(), q_expansion_modular_form(), GiNaC::relational::rhs(), and GiNaC::ex::series().

8.39.3.2 nops()

```
size_t GiNaC::Eisenstein_kernel::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.39.3.3 op()

```
ex GiNaC::Eisenstein_kernel::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References a, b, C_norm, K, k, and N.

8.39.3.4 let_op()

```
ex & GiNaC::Eisenstein_kernel::let_op (
            size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References a, b, C_norm, GiNaC::basic::ensure_if_modifiable(), K, k, and N.

8.39.3.5 is_numeric()

```
bool GiNaC::Eisenstein_kernel::is_numeric (
            void ) const  [override], [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented from GiNaC::integration_kernel.

References a, b, C_norm, GiNaC::ex::evalf(), GiNaC::ex::info(), GiNaC::info_flags::integer, K, k, N, GiNaC::info_flags::nonnegint, GiNaC::info_flags::numeric, and GiNaC::info_flags::posint.

8.39.3.6 Laurent_series()

```
ex GiNaC::Eisenstein_kernel::Laurent_series (
            const ex & x,
            int order) const  [override], [virtual]
```

Returns the Laurent series, starting possibly with the pole term.

Neglected terms are of order $O(x^order)$.

Reimplemented from GiNaC::integration_kernel.

References C_norm, q_expansion_modular_form(), and GiNaC::ex::series().

8.39.3.7 get_numerical_value()

```
ex GiNaC::Eisenstein_kernel::get_numerical_value (
            const ex & qbar,
            int N_trunc = 0) const  [override], [virtual]
```

Returns the value of the modular form.

Reimplemented from GiNaC::integration_kernel.

References C_norm, GiNaC::integration_kernel::get_numerical_value_impl(), and GiNaC::numeric::numeric().

8.39.3.8 uses_Laurent_series()

```
bool GiNaC::Eisenstein_kernel::uses_Laurent_series () const  [override], [protected], [virtual]
```

Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).

Returns false if this is not the case (and the class has an implementation of series_coeff_impl).

Reimplemented from GiNaC::integration_kernel.

8.39.3.9 q_expansion_modular_form()

```
ex GiNaC::Eisenstein_kernel::q_expansion_modular_form (
            const ex & q,
            int order) const
```

References a, b, GiNaC::ex_to(), K, k, and N.

Referenced by Laurent_series(), and series().

8.39.3.10 do_print()

```
void GiNaC::Eisenstein_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References a, b, C_norm, K, k, N, GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.39.4 Member Data Documentation

8.39.4.1 k

```
ex GiNaC::Eisenstein_kernel::k  [protected]
```

Referenced by do_print(), Eisenstein_kernel(), is_numeric(), let_op(), op(), and q_expansion_modular_form().

8.39.4.2 N

```
ex GiNaC::Eisenstein_kernel::N  [protected]
```

Referenced by do_print(), Eisenstein_kernel(), is_numeric(), let_op(), op(), and q_expansion_modular_form().

8.39.4.3 a

```
ex GiNaC::Eisenstein_kernel::a  [protected]
```

Referenced by do_print(), Eisenstein_kernel(), is_numeric(), let_op(), op(), and q_expansion_modular_form().

8.39.4.4 b

```
ex GiNaC::Eisenstein_kernel::b  [protected]
```

Referenced by do_print(), Eisenstein_kernel(), is_numeric(), let_op(), op(), and q_expansion_modular_form().

8.39.4.5 K

```
ex GiNaC::Eisenstein_kernel::K  [protected]
```

Referenced by do_print(), Eisenstein_kernel(), is_numeric(), let_op(), op(), and q_expansion_modular_form().

8.39.4.6 C_norm

```
ex GiNaC::Eisenstein_kernel::C_norm  [protected]
```

Referenced by do_print(), Eisenstein_kernel(), get_numerical_value(), is_numeric(), Laurent_series(), let_op(), and op().

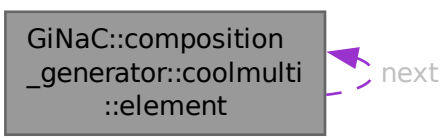
The documentation for this class was generated from the following files:

- integration_kernel.h
- integration_kernel.cpp

8.40 GiNaC::composition_generator::coolmulti::element Struct Reference

```
#include <utils.h>
```

Collaboration diagram for GiNaC::composition_generator::coolmulti::element:



Public Member Functions

- element (unsigned val, element ∗n)
- ∼element ()

Public Attributes

- unsigned value
- element ∗ next

8.40.1 Constructor & Destructor Documentation

8.40.1.1 element()

```
GiNaC::composition_generator::coolmulti::element::element (
            unsigned val,
            element * n)  [inline]
```

References element(), next, and value.

Referenced by GiNaC::composition_generator::coolmulti::coolmulti(), and element().

8.40.1.2 ∼element()

```
GiNaC::composition_generator::coolmulti::element::~element ()  [inline]
```

References next.

8.40.2 Member Data Documentation

8.40.2.1 value

```
unsigned GiNaC::composition_generator::coolmulti::element::value
```

Referenced by element(), GiNaC::composition_generator::coolmulti::finished(), GiNaC::composition_generator::get(), and GiNaC::composition_generator::coolmulti::next_permutation().

8.40.2.2 next

```
element* GiNaC::composition_generator::coolmulti::element::next
```

Referenced by GiNaC::composition_generator::coolmulti::coolmulti(), element(), GiNaC::composition_generator::coolmulti::finished(), GiNaC::composition_generator::get(), GiNaC::composition_generator::coolmulti::next_permutation(), and ∼element().

The documentation for this struct was generated from the following file:

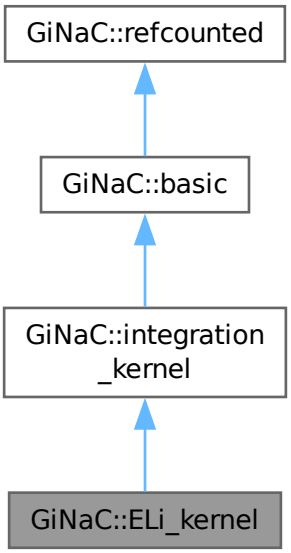
- utils.h

8.41 GiNaC::ELi_kernel Class Reference

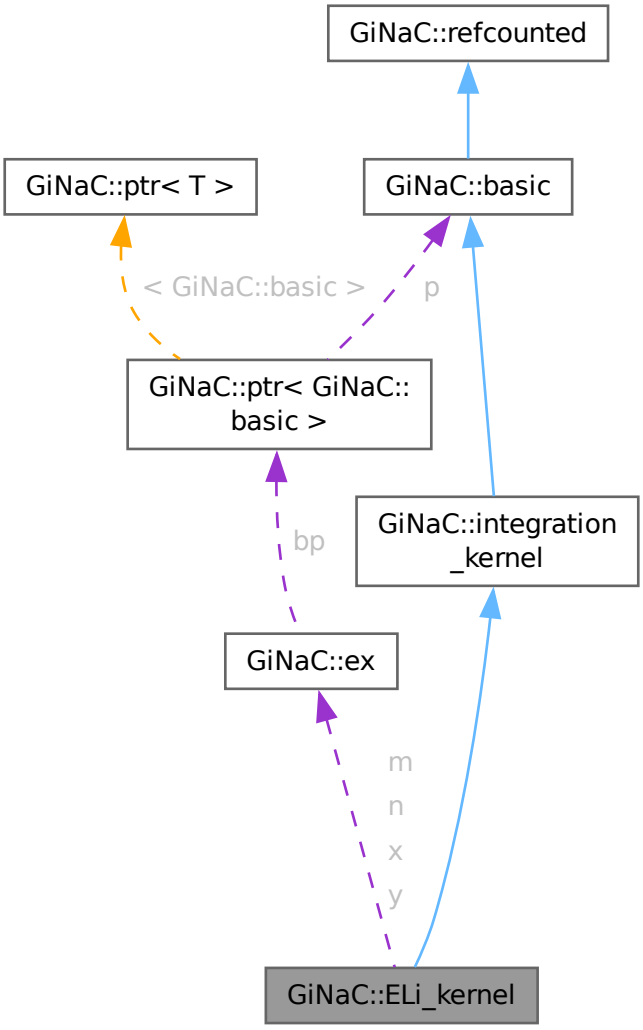
The ELi-kernel.

`#include <integration_kernel.h>`

Inheritance diagram for GiNaC::ELi_kernel:



Collaboration diagram for GiNaC::ELi_kernel:



Public Member Functions

- ELi_kernel (const ex &n, const ex &m, const ex &x, const ex &y)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- bool is_numeric (void) const override

  This routine returns true, if the integration kernel can be evaluated numerically.
- ex get_numerical_value (const ex &qbar, int N_trunc=0) const override

  Returns the value of ELi_{n,m}(x,y,qbar).

Public Member Functions inherited from GiNaC::integration_kernel

- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- virtual ex Laurent_series (const ex &x, int order) const

  Returns the Laurent series, starting possibly with the pole term.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

Load (deserialize) the object from an archive node.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- cln::cl_N series_coeff_impl (int i) const override

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual bool uses_Laurent_series () const

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex n
- ex m
- ex x
- ex y

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.41.1 Detailed Description

The ELi-kernel.

This class represents the differential one-form

$$\omega_{n;m}^{\mathrm{ELi}}(x;y) = \mathrm{ELi}_{n;m}(x;y;\bar{q})\frac{d\bar{q}}{\bar{q}}$$

8.41.2 Constructor & Destructor Documentation

8.41.2.1 ELi_kernel()

```
GiNaC::ELi_kernel::ELi_kernel (
             const ex & n,
             const ex & m,
             const ex & x,
             const ex & y)
```

References ELi_kernel(), m, n, x, and y.

Referenced by ELi_kernel().

8.41.3 Member Function Documentation

8.41.3.1 nops()

```
size_t GiNaC::ELi_kernel::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.41.3.2 op()

```
ex GiNaC::ELi_kernel::op (
             size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References m, n, x, and y.

8.41.3.3 let_op()

```
ex & GiNaC::ELi_kernel::let_op (
             size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ensure_if_modifiable(), m, n, x, and y.

8.41.3.4 is_numeric()

```
bool GiNaC::ELi_kernel::is_numeric (
            void ) const  [override], [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented from GiNaC::integration_kernel.

References GiNaC::ex::evalf(), GiNaC::ex::info(), m, n, GiNaC::info_flags::nonnegint, GiNaC::info_flags::numeric, x, and y.

8.41.3.5 get_numerical_value()

```
ex GiNaC::ELi_kernel::get_numerical_value (
            const ex & qbar,
            int N_trunc = 0) const  [override], [virtual]
```

Returns the value of ELi_{n,m}(x,y,qbar).

Reimplemented from GiNaC::integration_kernel.

References GiNaC::integration_kernel::get_numerical_value_impl().

8.41.3.6 series_coeff_impl()

```
cln::cl_N GiNaC::ELi_kernel::series_coeff_impl (
            int i) const  [override], [protected], [virtual]
```

For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.

The i-th coefficient corresponds to the power λ^{i-1}.

Reimplemented from GiNaC::integration_kernel.

References GiNaC::ex::evalf(), GiNaC::ex_to(), m, n, x, and y.

8.41.3.7 do_print()

```
void GiNaC::ELi_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References m, n, GiNaC::ex::print(), GiNaC::print_context::s, x, and y.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.41.4 Member Data Documentation

8.41.4.1 n

```
ex GiNaC::ELi_kernel::n  [protected]
```

Referenced by do_print(), ELi_kernel(), is_numeric(), let_op(), op(), and series_coeff_impl().

8.41.4.2 m

```
ex GiNaC::ELi_kernel::m  [protected]
```

Referenced by do_print(), ELi_kernel(), is_numeric(), let_op(), op(), and series_coeff_impl().

8.41.4.3 x

```
ex GiNaC::ELi_kernel::x  [protected]
```

Referenced by do_print(), ELi_kernel(), is_numeric(), let_op(), op(), and series_coeff_impl().

8.41.4.4 y

```
ex GiNaC::ELi_kernel::y  [protected]
```

Referenced by do_print(), ELi_kernel(), is_numeric(), let_op(), op(), and series_coeff_impl().

The documentation for this class was generated from the following files:

- integration_kernel.h
- integration_kernel.cpp

8.42 std::equal_to< GiNaC::ex > Struct Reference

Specialization of std::equal_to() for ex objects.

```
#include <ex.h>
```

Public Member Functions

- bool operator() (const GiNaC::ex &e1, const GiNaC::ex &e2) const noexcept

8.42.1 Detailed Description

Specialization of std::equal_to() for ex objects.

8.42.2 Member Function Documentation

8.42.2.1 operator()()

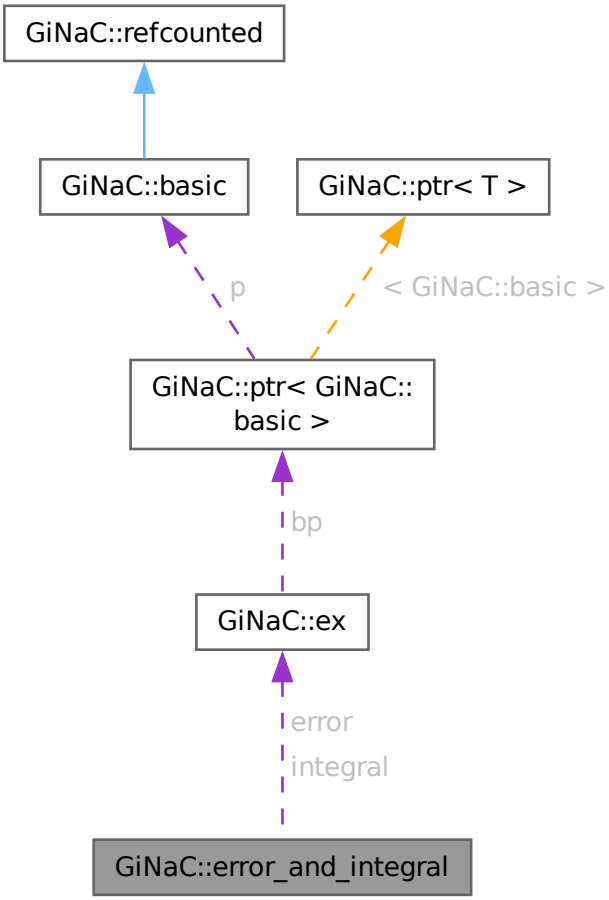
```
bool std::equal_to< GiNaC::ex >::operator() (
            const GiNaC::ex & e1,
            const GiNaC::ex & e2) const  [inline], [noexcept]
```

The documentation for this struct was generated from the following file:

- ex.h

8.43 GiNaC::error_and_integral Struct Reference

Collaboration diagram for GiNaC::error_and_integral:



Public Member Functions

- error_and_integral (const ex &err, const ex &integ)

Public Attributes

- ex error
- ex integral

8.43.1 Constructor & Destructor Documentation

8.43.1.1 error_and_integral()

```
GiNaC::error_and_integral::error_and_integral (
            const ex & err,
            const ex & integ)  [inline]
```

References error, and integral.

Referenced by GiNaC::adaptivesimpson().

8.43.2 Member Data Documentation

8.43.2.1 error

```
ex GiNaC::error_and_integral::error
```

Referenced by error_and_integral(), and GiNaC::error_and_integral_is_less::operator()().

8.43.2.2 integral

```
ex GiNaC::error_and_integral::integral
```

Referenced by error_and_integral(), and GiNaC::error_and_integral_is_less::operator()().

The documentation for this struct was generated from the following file:

- integral.cpp

8.44 GiNaC::error_and_integral_is_less Struct Reference

Public Member Functions

- bool operator() (const error_and_integral &e1, const error_and_integral &e2) const

8.44.1 Member Function Documentation

8.44.1.1 operator()()

```
bool GiNaC::error_and_integral_is_less::operator() (
            const error_and_integral & e1,
            const error_and_integral & e2) const  [inline]
```

References GiNaC::ex::compare(), GiNaC::error_and_integral::error, and GiNaC::error_and_integral::integral.

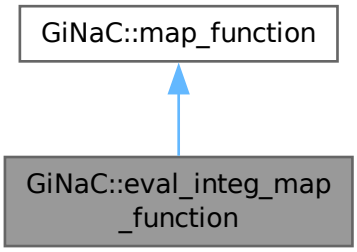
The documentation for this struct was generated from the following file:

- integral.cpp

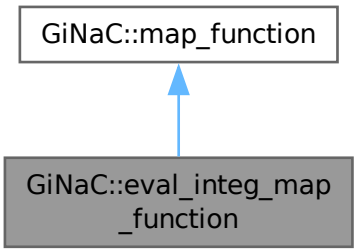
8.45 GiNaC::eval_integ_map_function Struct Reference

Function object to be applied by basic::eval_integ().

Inheritance diagram for GiNaC::eval_integ_map_function:



Collaboration diagram for GiNaC::eval_integ_map_function:



Public Member Functions

- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.45.1 Detailed Description

Function object to be applied by basic::eval_integ().

8.45.2 Member Function Documentation

8.45.2.1 operator()()

```
ex GiNaC::eval_integ_map_function::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References GiNaC::eval_integ().

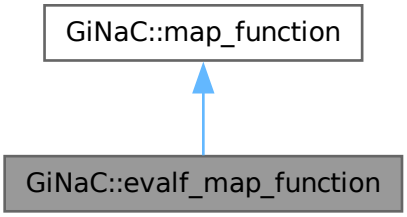
The documentation for this struct was generated from the following file:

- basic.cpp

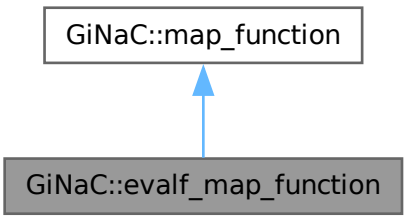
8.46 GiNaC::evalf_map_function Struct Reference

Function object to be applied by basic::evalf().

Inheritance diagram for GiNaC::evalf_map_function:



Collaboration diagram for GiNaC::evalf_map_function:



Public Member Functions

- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.46.1 Detailed Description

Function object to be applied by basic::evalf().

8.46.2 Member Function Documentation

8.46.2.1 operator()()

```
ex GiNaC::evalf_map_function::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References GiNaC::evalf().

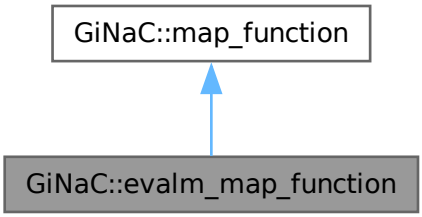
The documentation for this struct was generated from the following file:

- basic.cpp

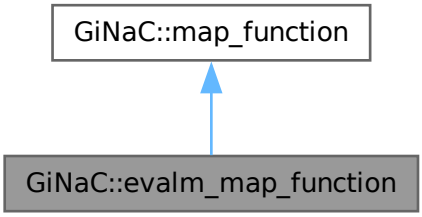
8.47 GiNaC::evalm_map_function Struct Reference

Function object to be applied by basic::evalm().

Inheritance diagram for GiNaC::evalm_map_function:



Collaboration diagram for GiNaC::evalm_map_function:



Public Member Functions

- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.47.1 Detailed Description

Function object to be applied by basic::evalm().

8.47.2 Member Function Documentation

8.47.2.1 operator()()

```
ex GiNaC::evalm_map_function::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References GiNaC::evalm().

The documentation for this struct was generated from the following file:

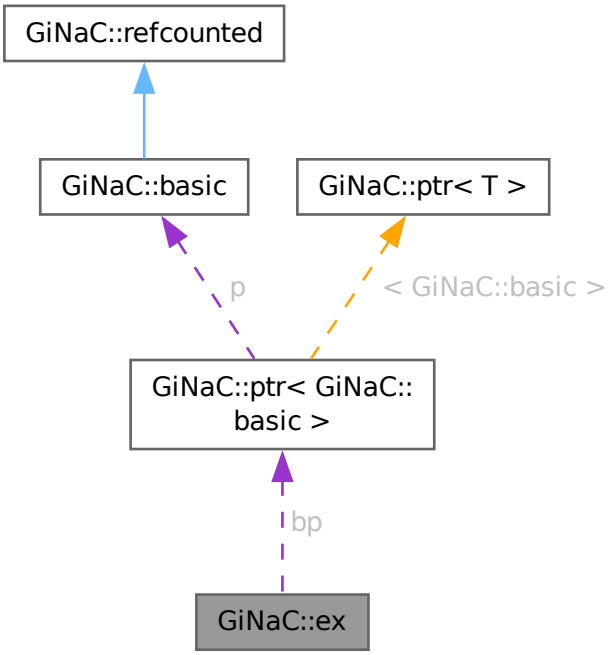
- basic.cpp

8.48 GiNaC::ex Class Reference

Lightweight wrapper for GiNaC's symbolic objects.

```
#include <ex.h>
```

Collaboration diagram for GiNaC::ex:



Public Member Functions

- ex () noexcept
- ex (const basic &other)
- ex (int i)
- ex (unsigned int i)
- ex (long i)
- ex (unsigned long i)
- ex (long long i)
- ex (unsigned long long i)
- ex (double const d)
- ex (const std::string &s, const ex &l)

  Construct ex from string and a list of symbols.
- void swap (ex &other) noexcept

  Efficiently swap the contents of two expressions.
- const_iterator begin () const noexcept
- const_iterator end () const noexcept
- const_preorder_iterator preorder_begin () const
- const_preorder_iterator preorder_end () const noexcept
- const_postorder_iterator postorder_begin () const
- const_postorder_iterator postorder_end () const noexcept
- ex eval () const
- ex evalf () const
- ex evalm () const
- ex eval_ncmul (const exvector &v) const
- ex eval_integ () const
- void print (const print_context &c, unsigned level=0) const

  Print expression to stream.
- void dbgprint () const

  Little wrapper arount print to be called within a debugger.
- void dbgprinttree () const

  Little wrapper arount printtree to be called within a debugger.
- bool info (unsigned inf) const
- size_t nops () const
- ex op (size_t i) const
- ex operator[ ] (const ex &index) const
- ex operator[ ] (size_t i) const
- ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- ex & operator[ ] (const ex &index)
- ex & operator[ ] (size_t i)
- ex lhs () const

  Left hand side of relational expression.
- ex rhs () const

  Right hand side of relational expression.
- ex conjugate () const
- ex real_part () const
- ex imag_part () const
- bool has (const ex &pattern, unsigned options=0) const
- bool find (const ex &pattern, exset &found) const

  Find all occurrences of a pattern.
- bool match (const ex &pattern) const

  Check whether expression matches a specified pattern.

- bool match (const ex &pattern, exmap &repls) const
- ex subs (const exmap &m, unsigned options=0) const
- ex subs (const lst &ls, const lst &lr, unsigned options=0) const

  Substitute objects in an expression (syntactic substitution) and return the result as a new expression.
- ex subs (const ex &e, unsigned options=0) const

  Substitute objects in an expression (syntactic substitution) and return the result as a new expression.
- ex map (map_function &f) const
- ex map (ex(∗f)(const ex &e)) const
- void accept (visitor &v) const
- void traverse_preorder (visitor &v) const

  Traverse expression tree with given visitor, preorder traversal.
- void traverse_postorder (visitor &v) const

  Traverse expression tree with given visitor, postorder traversal.
- void traverse (visitor &v) const
- bool is_polynomial (const ex &vars) const

  Check whether expression is a polynomial.
- int degree (const ex &s) const
- int ldegree (const ex &s) const
- ex coeff (const ex &s, int n=1) const
- ex lcoeff (const ex &s) const
- ex tcoeff (const ex &s) const
- ex expand (unsigned options=0) const

  Expand an expression.
- ex collect (const ex &s, bool distributed=false) const
- ex diff (const symbol &s, unsigned nth=1) const

  Compute partial derivative of an expression.
- ex series (const ex &r, int order, unsigned options=0) const

  Compute the truncated series expansion of an expression.
- ex normal () const

  Normalization of rational functions.
- ex to_rational (exmap &repl) const

  Rationalization of non-rational functions.
- ex to_polynomial (exmap &repl) const
- ex numer () const

  Get numerator of an expression.
- ex denom () const

  Get denominator of an expression.
- ex numer_denom () const

  Get numerator and denominator of an expression.
- ex unit (const ex &x) const

  Compute unit part (= sign of leading coefficient) of a multivariate polynomial in Q[x].
- ex content (const ex &x) const

  Compute content part (= unit normal GCD of all coefficients) of a multivariate polynomial in Q[x].
- numeric integer_content () const

  Compute the integer content (= GCD of all numeric coefficients) of an expanded polynomial.
- ex primpart (const ex &x) const

  Compute primitive part of a multivariate polynomial in Q[x].
- ex primpart (const ex &x, const ex &cont) const

  Compute primitive part of a multivariate polynomial in Q[x] when the content part is already known.
- void unitcontprim (const ex &x, ex &u, ex &c, ex &p) const

  Compute unit part, content part, and primitive part of a multivariate polynomial in Q[x].
- ex smod (const numeric &xi) const

- numeric max_coefficient () const

  Return maximum (absolute value) coefficient of a polynomial.
- exvector get_free_indices () const
- ex simplify_indexed (unsigned options=0) const

  Simplify/canonicalize expression containing indexed objects.
- ex simplify_indexed (const scalar_products &sp, unsigned options=0) const

  Simplify/canonicalize expression containing indexed objects.
- int compare (const ex &other) const
- bool is_equal (const ex &other) const
- bool is_zero () const
- bool is_zero_matrix () const

  Check whether expression is zero or zero matrix.
- ex symmetrize () const

  Symmetrize expression over its free indices.
- ex symmetrize (const lst &l) const

  Symmetrize expression over a list of objects (symbols, indices).
- ex antisymmetrize () const

  Antisymmetrize expression over its free indices.
- ex antisymmetrize (const lst &l) const

  Antisymmetrize expression over a list of objects (symbols, indices).
- ex symmetrize_cyclic () const

  Symmetrize expression by cyclic permutation over its free indices.
- ex symmetrize_cyclic (const lst &l) const

  Symmetrize expression by cyclic permutation over a list of objects (symbols, indices).
- unsigned return_type () const
- return_type_t return_type_tinfo () const
- unsigned gethash () const

Private Member Functions

- void makewriteable ()

  Make this ex writable (if more than one ex handle the same basic) by unlinking the object and creating an unshared copy of it.
- void share (const ex &other) const

  Share equal objects between expressions.

Static Private Member Functions

- static ptr< basic > construct_from_basic (const basic &other)

  Helper function for the ex-from-basic constructor.
- static basic & construct_from_int (int i)
- static basic & construct_from_uint (unsigned int i)
- static basic & construct_from_long (long i)
- static basic & construct_from_ulong (unsigned long i)
- static basic & construct_from_longlong (long long i)
- static basic & construct_from_ulonglong (unsigned long long i)
- static basic & construct_from_double (double d)
- static ptr< basic > construct_from_string_and_lst (const std::string &s, const ex &l)

Private Attributes

- ptr< basic > bp

  pointer to basic object managed by this

Friends

- class archive_node
- bool are_ex_trivially_equal (const ex &e1, const ex &e2)

  Compare two objects of class quickly without doing a deep tree traversal.
- template<class T>
  const T & ex_to (const ex &e)

  Return a reference to the basic-derived class T object embedded in an expression.
- template<class T>
  bool is_a (const ex &obj)

  Check if ex is a handle to a T, including base classes.
- template<class T>
  bool is_exactly_a (const ex &obj)

  Check if ex is a handle to a T, not including base classes.

8.48.1 Detailed Description

Lightweight wrapper for GiNaC's symbolic objects.

It holds a pointer to the other object in order to do garbage collection by the method of reference counting. I.e., it is a smart pointer. Also, the constructor ex::ex(const basic & other) calls the methods that do automatic evaluation. E.g., x-x turns automatically into 0.

8.48.2 Constructor & Destructor Documentation

8.48.2.1 ex() [1/10]

```
GiNaC::ex::ex ()  [inline], [noexcept]
```

References GiNaC::_num0_bp, bp, GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

Referenced by antisymmetrize(), antisymmetrize(), archive_node, are_ex_trivially_equal, GiNaC::matrix::charpoly(), coeff(), collect(), compare(), conjugate(), construct_from_basic(), construct_from_string_and_lst(), content(), degree(), denom(), diff(), GiNaC::mul::do_print_csrc(), eval(), GiNaC::function::eval(), GiNaC::indexed::eval(), eval_integ(), eval_ncmul(), evalf(), evalm(), ex(), ex_to, GiNaC::exp_eval(), expand(), GiNaC::integral::expand(), GiNaC::power::expand_mul(), find(), GiNaC::G(), GiNaC::G(), has(), imag_part(), is_a, is_equal(), is_exactly_a, is_polynomial(), is_zero(), is_zero_matrix(), GiNaC::iterated_integral(), GiNaC::iterated_integral(), lcoeff(), ldegree(), let_op(), lhs(), GiNaC::library_init::library_init(), map(), map(), match(), match(), normal(), numer(), numer_denom(), op(), GiNaC::const_iterator::operator->(), operator[](), operator[](), operator[](), operator[](), GiNaC::pow(), primpart(), primpart(), GiNaC::psi(), GiNaC::psi(), GiNaC::psi2_eval(), real_part(), rhs(), GiNaC::structure< T, ComparisonPolicy >::scalar_mul_indexed(), series(), share(), simplify_indexed(), simplify_indexed(), GiNaC::indexed::simplify_indexed_product, smod(), subs(), subs(), subs(), swap(), symmetrize(), symmetrize(), symmetrize_cyclic(), symmetrize_cyclic(), tcoeff(), to_polynomial(), to_rational(), GiNaC::archive_node::unarchive(), unit(), unitcontprim(), GiNaC::zeta(), and GiNaC::zeta().

8.48.2.2 ex() [2/10]

```
GiNaC::ex::ex (
            const basic & other)  [inline]
```

References bp, construct_from_basic(), GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

8.48.2.3 ex() [3/10]

```
GiNaC::ex::ex (
            int i)  [inline]
```

References bp, construct_from_int(), GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

8.48.2.4 ex() [4/10]

```
GiNaC::ex::ex (
            unsigned int i)  [inline]
```

References bp, construct_from_uint(), GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

8.48.2.5 ex() [5/10]

```
GiNaC::ex::ex (
            long i)  [inline]
```

References bp, construct_from_long(), GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

8.48.2.6 ex() [6/10]

```
GiNaC::ex::ex (
            unsigned long i)  [inline]
```

References bp, construct_from_ulong(), GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

8.48.2.7 ex() [7/10]

```
GiNaC::ex::ex (
            long long i)  [inline]
```

References bp, construct_from_longlong(), GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

8.48.2.8 ex() [8/10]

```
GiNaC::ex::ex (
            unsigned long long i)  [inline]
```

References bp, construct_from_ulonglong(), GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

8.48.2.9 ex() [9/10]

```
GiNaC::ex::ex (
            double const d)  [inline]
```

References bp, construct_from_double(), GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

8.48.2.10 ex() [10/10]

```
GiNaC::ex::ex (
            const std::string & s,
            const ex & l)  [inline]
```

Construct ex from string and a list of symbols.

The input grammar is similar to the GiNaC output format. All symbols and indices to be used in the expression must be specified in a lst in the second argument. Undefined symbols and other parser errors will throw an exception.

References bp, construct_from_string_and_lst(), GiNaC::status_flags::dynallocated, ex(), and GINAC_ASSERT.

8.48.3 Member Function Documentation

8.48.3.1 swap()

```
void GiNaC::ex::swap (
            ex & other)  [inline], [noexcept]
```

Efficiently swap the contents of two expressions.

References bp, GiNaC::status_flags::dynallocated, ex(), GINAC_ASSERT, and swap().

Referenced by GiNaC::ncmul::derivative(), GiNaC::ex_swap::operator()(), swap(), GiNaC::expair::swap(), GiNaC::swap(), and std::swap().

8.48.3.2 begin()

```
const_iterator GiNaC::ex::begin () const  [inline], [noexcept]
```

References GiNaC::const_iterator::const_iterator().

Referenced by GiNaC::abs_expand(), GiNaC::antisymmetrize(), GiNaC::symmetry::canonicalize, GiNaC::exp_expand(), GiNaC::log_expand(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::symmetrize(), and GiNaC::symmetrize_cyclic().

8.48.3.3 end()

```
const_iterator GiNaC::ex::end () const  [inline], [noexcept]
```

References GiNaC::const_iterator::const_iterator(), and nops().

Referenced by GiNaC::abs_expand(), GiNaC::symmetry::canonicalize, GiNaC::exp_expand(), GiNaC::log_expand(), and GiNaC::rename_dummy_indices_uniquely().

8.48.3.4 preorder_begin()

```
const_preorder_iterator GiNaC::ex::preorder_begin () const  [inline]
```

References GiNaC::const_preorder_iterator::const_preorder_iterator(), and nops().

8.48.3.5 preorder_end()

```
const_preorder_iterator GiNaC::ex::preorder_end () const  [inline], [noexcept]
```

References GiNaC::const_preorder_iterator::const_preorder_iterator().

8.48.3.6 postorder_begin()

```
const_postorder_iterator GiNaC::ex::postorder_begin () const  [inline]
```

References GiNaC::const_postorder_iterator::const_postorder_iterator(), and nops().

8.48.3.7 postorder_end()

```
const_postorder_iterator GiNaC::ex::postorder_end () const  [inline], [noexcept]
```

References GiNaC::const_postorder_iterator::const_postorder_iterator().

8.48.3.8 eval()

```
ex GiNaC::ex::eval () const  [inline]
```

References bp, eval(), and ex().

Referenced by GiNaC::eval(), and eval().

8.48.3.9 evalf()

```
ex GiNaC::ex::evalf () const  [inline]
```

References bp, and ex().

Referenced by GiNaC::adaptivesimpson(), GiNaC::EllipticE_evalf(), GiNaC::EllipticK_evalf(), GiNaC::constant::evalf(), GiNaC::evalf(), GiNaC::Gt::evalf(), GiNaC::integral::evalf(), GiNaC::mul::evalf(), GiNaC::power::evalf(), GiNaC::fsolve(), GiNaC::Kronecker_dtau_kernel::get_numerical_value(), GiNaC::integration_kernel::get_numerical_value_impl(), GiNaC::Gt::Gt(), GiNaC::Gt::Gt(), GiNaC::H_evalf(), GiNaC::Ebar_kernel::is_numeric(), GiNaC::Eisenstein_h_kernel::is_numeric(), GiNaC::Eisenstein_kernel::is_numeric(), GiNaC::ELi_kernel::is_numeric(), GiNaC::Kronecker_dtau_kernel::is_numeric(), GiNaC::Kronecker_dz_kernel::is_numeric(), GiNaC::modular_form_kernel::is_numeric(), GiNaC::multiple_polylog_kernel::is_numeric GiNaC::iterated_integral_evalf_impl(), GiNaC::Li_evalf(), GiNaC::S_evalf(), GiNaC::integration_kernel::series_coeff(), GiNaC::Ebar_kernel::series_coeff_impl(), GiNaC::ELi_kernel::series_coeff_impl(), GiNaC::Kronecker_dtau_kernel::series_coeff_impl GiNaC::Kronecker_dz_kernel::series_coeff_impl(), GiNaC::multiple_polylog_kernel::series_coeff_impl(), and GiNaC::subsvalue().

8.48.3.10 evalm()

```
ex GiNaC::ex::evalm () const  [inline]
```

References bp, and ex().

Referenced by GiNaC::add::evalm(), GiNaC::evalm(), GiNaC::mul::evalm(), GiNaC::power::evalm(), and is_zero_matrix().

8.48.3.11 eval_ncmul()

```
ex GiNaC::ex::eval_ncmul (
            const exvector & v) const  [inline]
```

References bp, and ex().

Referenced by GiNaC::integral::eval_ncmul(), and GiNaC::relational::eval_ncmul().

8.48.3.12 eval_integ()

```
ex GiNaC::ex::eval_integ () const  [inline]
```

References bp, and ex().

Referenced by GiNaC::eval_integ(), GiNaC::integral::eval_integ(), and GiNaC::pseries::eval_integ().

8.48.3.13 print()

```
void GiNaC::ex::print (
            const print_context & c,
            unsigned level = 0) const
```

Print expression to stream.

The formatting of the output is determined by the kind of print_context object that is passed. Possible formattings include ginsh-parsable output (the default), tree-like output for debugging, and C++ source.

See also

print_context

References bp.

Referenced by GiNaC::abs_print_csrc_float(), GiNaC::abs_print_latex(), GiNaC::conjugate_print_latex(), GiNaC::Ebar_kernel::do_print(), GiNaC::Eisenstein_h_kernel::do_print(), GiNaC::Eisenstein_kernel::do_print(), GiNaC::ELi_kernel::do_print(), GiNaC::integral::do_print(), GiNaC::Kronecker_dtau_kernel::do_print(), GiNaC::Kronecker_dz_kernel::do_print(), GiNaC::modular_form_kernel::do_print(), GiNaC::mul::do_print(), GiNaC::multiple_polylog_kernel::do_print(), GiNaC::relational::do_print(), GiNaC::user_defined_kernel::do_print(), GiNaC::add::do_print_csrc(), GiNaC::idx::do_print_csrc(), GiNaC::mul::do_print_csrc(), GiNaC::power::do_print_csrc(), GiNaC::power::do_print_csrc_cl_N(), GiNaC::power::do_print_dflt(), GiNaC::integral::do_print_latex(), GiNaC::power::do_print_latex(), GiNaC::add::do_print_python_repr(), GiNaC::mul::do_print_python_repr(), GiNaC::power::do_print_python_repr(), GiNaC::pseries::do_print_python_repr(), GiNaC::relational::do_print_python_repr(), GiNaC::basic::do_print_tree(), GiNaC::clifford::do_print_tree(), GiNaC::expairseq::do_print_tree(), GiNaC::idx::do_print_tree(), GiNaC::pseries::do_print_tree(), GiNaC::spinidx::do_print_tree(), GiNaC::varidx::do_print_tree(), GiNaC::EllipticE_print_latex(), GiNaC::EllipticK_print_latex(), GiNaC::factorial_print_dflt_latex(), GiNaC::H_print_latex(), GiNaC::imag_part_print_latex(), GiNaC::operator<<(), GiNaC::expair::print(), GiNaC::add::print_add(), GiNaC::idx::print_index(), GiNaC::mul::print_overall_coeff(), GiNaC::power::print_power(), GiNaC::pseries::print_series(), GiNaC::expairseq::printpair(), GiNaC::expairseq::printseq(), GiNaC::real_part_print_latex(), GiNaC::S_print_latex(), and GiNaC::zeta1_print_latex().

8.48.3.14 dbgprint()

```
void GiNaC::ex::dbgprint () const
```

Little wrapper arount print to be called within a debugger.

References bp.

8.48.3.15 dbgprinttree()

```
void GiNaC::ex::dbgprinttree () const
```

Little wrapper arount printtree to be called within a debugger.

References bp.

8.48.3.16 info()

```
bool GiNaC::ex::info (
            unsigned inf) const  [inline]
```

References bp.

Referenced by GiNaC::abs_eval(), GiNaC::abs_info(), GiNaC::abs_power(), GiNaC::acos_eval(), GiNaC::acosh_eval(), GiNaC::asin_eval(), GiNaC::asin_info(), GiNaC::asinh_conjugate(), GiNaC::asinh_eval(), GiNaC::atan2_eval(), GiNaC::atan2_info(), GiNaC::atan_conjugate(), GiNaC::atan_eval(), GiNaC::atan_info(), GiNaC::atanh_eval(), GiNaC::beta_eval(), GiNaC::beta_series(), GiNaC::mul::can_be_further_expanded(), GiNaC::mul::can_make_flat(), GiNaC::canonicalize_clifford(), GiNaC::clifford_moebius_map(), GiNaC::clifford_prime(), GiNaC::clifford_star_bar(), GiNaC::power::conjugate(), GiNaC::pseries::conjugate(), GiNaC::expairseq::construct_from_2_ex(), content(), GiNaC::cos_eval(), GiNaC::cosh_eval(), GiNaC::csgn_series(), GiNaC::matrix::determinant(), GiNaC::divide(), GiNaC::divide_in_z(), GiNaC::add::do_print_csrc(), GiNaC::idx::do_print_csrc(), GiNaC::power::do_print_csrc(), GiNaC::EllipticE_eval(), GiNaC::EllipticE_evalf(), GiNaC::EllipticK_eval(), GiNaC::EllipticK_evalf(), GiNaC::eta_eval(), GiNaC::eta_evalf(), GiNaC::eta_series(), GiNaC::power::eval(), GiNaC::exp_eval(), GiNaC::exp_info(), GiNaC::exp_power(), GiNaC::power::expand(), GiNaC::find_common_factor(), GiNaC::func_arg_info(), GiNaC::G2_eval(), GiNaC::G2_evalf(), GiNaC::G3_eval(), GiNaC::G3_evalf(), GiNaC::gcd(), GiNaC::get_clifford_comp(), GiNaC::get_dim_uint(), GiNaC::H_eval(), GiNaC::H_evalf(), GiNaC::power::has(), GiNaC::heur_gcd(), GiNaC::idx::idx(), GiNaC::power::imag_part(), GiNaC::pseries::imag_part(), GiNaC::add::info(), GiNaC::mul::info(), GiNaC::power::info(), GiNaC::Ebar_kernel::is_numeric(), GiNaC::Eisenstein_h_kernel::is_numeric(), GiNaC::Eisenstein_kernel::is_numeric(), GiNaC::ELi_kernel::is_numeric(), GiNaC::Kronecker_dtau_kernel::is_numeric(), GiNaC::Kronecker_dz_kernel::is_numeric(), GiNaC::modular_form_kernel::is_numeric(GiNaC::multiple_polylog_kernel::is_numeric(), GiNaC::power::is_polynomial(), GiNaC::iterated_integral2_eval(), GiNaC::iterated_integral3_eval(), GiNaC::iterated_integral_evalf_impl(), GiNaC::lcm(), GiNaC::lcmcoeff(), GiNaC::lgamma_conjugate(), GiNaC::lgamma_eval(), GiNaC::lgamma_series(), GiNaC::Li2_conjugate(), GiNaC::Li2_eval(), GiNaC::Li2_series(), GiNaC::Li_eval(), GiNaC::Li_evalf(), GiNaC::Li_series(), GiNaC::log_conjugate(), GiNaC::log_eval(), GiNaC::log_expand(), GiNaC::log_imag_part(), GiNaC::log_info(), GiNaC::log_real_part(), GiNaC::log_series(), GiNaC::lsolve(), GiNaC::lst_to_clifford(), GiNaC::basic::normal(), GiNaC::power::normal(), GiNaC::relational::operator safe_bool(), GiNaC::Order_imag_part(), GiNaC::Order_power(), GiNaC::matrix::pow(), GiNaC::prem(), GiNaC::psi1_eval(), GiNaC::psi1_series(), GiNaC::psi2_eval(), GiNaC::psi2_series(), GiNaC::quo(), GiNaC::power::real_part(), GiNaC::pseries::real_part(), GiNaC::rem(), GiNaC::remove_dirac_ONE(), GiNaC::resultant(), GiNaC::S_eval(), GiNaC::S_evalf(), GiNaC::S_series(), GiNaC::mul::series(), GiNaC::power::series(), GiNaC::sin_eval(), GiNaC::sinh_eval(), GiNaC::sprem(), GiNaC::step_series(), subs(), GiNaC::tan_eval(), GiNaC::tanh_eval(), GiNaC::tgamma_eval(), GiNaC::tgamma_series(), GiNaC::expairseq::to_polynomial(), GiNaC::power::to_polynomial(), GiNaC::expairseq::to_rational(), GiNaC::power::to_rational(), GiNaC::matrix::trace(), GiNaC::trig_info(), GiNaC::tryfactsubs(), unit(), unitcontprim(), GiNaC::zeta1_eval(), GiNaC::zeta2_deriv(), GiNaC::zeta2_eval(), and GiNaC::zetaderiv_eval().

8.48.3.17 nops()

```
size_t GiNaC::ex::nops () const  [inline]
```

References bp.

Referenced by GiNaC::abs_expand(), GiNaC::matrix::add_indexed(), GiNaC::algebraic_match_mul_with_mul(), GiNaC::ncmul::append_factors(), GiNaC::canonicalize_clifford(), GiNaC::clifford_max_label(), GiNaC::clifford_star_bar(), GiNaC::basic::collect(), GiNaC::collect_symbols(), GiNaC::color_trace(), GiNaC::ncmul::count_factors(), GiNaC::csgn_eval(), GiNaC::const_postorder_iterator::descend(), GiNaC::dirac_trace(), GiNaC::divide(), end(), GiNaC::indexed::eval(), GiNaC::exp_expand(), GiNaC::indexed::expand(), GiNaC::integral::expand(), GiNaC::mul::expand(), find(), GiNaC::find_common_factor(), GiNaC::gcd_pf_mul(), GiNaC::get_all_dummy_indices_safely(), GiNaC::get_clifford_comp(), GiNaC::get_first_symbol(), GiNaC::get_symbol_stats(), GiNaC::H_evalf(), GiNaC::hasindex(), GiNaC::haswild(), GiNaC::idx_symmetrization(), GiNaC::Gt_detail::TransformExpressionWithCache< Type >::impl(), GiNaC::const_preorder_iterator::i GiNaC::lcmcoeff(), GiNaC::Li2_series(), GiNaC::Li_deriv(), GiNaC::log_expand(), GiNaC::lsolve(), GiNaC::basic::match(), GiNaC::expairseq::match(), GiNaC::multiply_lcm(), GiNaC::nops(), postorder_begin(), preorder_begin(), GiNaC::product_to_exvector(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::indexed::reposition_dummy_indices, GiNaC::matrix::scalar_mul_indexed(), GiNaC::integral::series(), GiNaC::Gt::setArgs(), GiNaC::indexed::simplify_indexed, GiNaC::indexed::simplify_indexed_product, GiNaC::step_eval(), GiNaC::symminfo::symminfo(), traverse_postorder(), traverse_preorder(), GiNaC::zeta1_eval(), GiNaC::zeta1_evalf(), and GiNaC::zeta2_evalf().

8.48.3.18 op()

```
ex GiNaC::ex::op (
            size_t i) const  [inline]
```

References bp, and ex().

Referenced by GiNaC::abs_eval(), GiNaC::matrix::add_indexed(), GiNaC::algebraic_match_mul_with_mul(), GiNaC::ncmul::append_factors(), GiNaC::atan_series(), GiNaC::atanh_series(), GiNaC::mul::can_be_further_expanded(), GiNaC::canonicalize_clifford(), GiNaC::clifford_max_label(), GiNaC::clifford_moebius_map(), GiNaC::clifford_prime(), GiNaC::clifford_star_bar(), GiNaC::clifford_to_lst(), GiNaC::basic::collect(), GiNaC::collect_symbols(), GiNaC::color_trace(), GiNaC::cliffordunit::contract_with(), GiNaC::matrix::contract_with(), GiNaC::cos_eval(), GiNaC::cosh_eval(), GiNaC::ncmul::count_factors(), GiNaC::csgn_eval(), GiNaC::decomp_rational(), denom(), GiNaC::const_postorder_iterator::descend(GiNaC::dirac_trace(), GiNaC::divide(), GiNaC::divide_in_z(), GiNaC::epsilon_tensor(), GiNaC::epsilon_tensor(), GiNaC::indexed::eval(), GiNaC::power::eval(), GiNaC::integral::eval_integ(), GiNaC::clifford::eval_ncmul(), GiNaC::exp_eval(), GiNaC::indexed::expand(), GiNaC::integral::expand(), GiNaC::mul::expand(), find(), GiNaC::find_common_factor() GiNaC::matrix::fraction_free_elimination(), GiNaC::gcd_pf_mul(), GiNaC::gcd_pf_pow(), GiNaC::gcd_pf_pow_pow(), GiNaC::get_all_dummy_indices_safely(), GiNaC::get_clifford_comp(), GiNaC::get_first_symbol(), GiNaC::clifford::get_metric(), GiNaC::H_evalf(), GiNaC::power::has(), GiNaC::hasindex(), GiNaC::haswild(), GiNaC::Gt_detail::TransformExpressionWithCache< T GiNaC::const_preorder_iterator::increment(), GiNaC::lcmcoeff(), GiNaC::Li2_series(), GiNaC::Li_deriv(), GiNaC::log_eval(), GiNaC::lorentz_eps(), GiNaC::lsolve(), GiNaC::lst_to_clifford(), GiNaC::basic::match(), GiNaC::expairseq::match(), GiNaC::multiply_lcm(), GiNaC::add::normal(), GiNaC::basic::normal(), normal(), GiNaC::mul::normal(), GiNaC::power::normal(), numer(), GiNaC::op(), GiNaC::ex_base_is_less::operator()(), GiNaC::op0_is_equal::operator()(), GiNaC::const_iterator::operator*(), GiNaC::const_iterator::operator[ ](), GiNaC::product_to_exvector(), GiNaC::remove_dirac_ONE(), GiNaC::replace_with_symbol(), GiNaC::indexed::reposition_dummy_indices, GiNaC::clifford::same_metric(), GiNaC::matrix::scalar_mul_indexed(), GiNaC::Gt::setArgs(), GiNaC::indexed::simplify_indexed, GiNaC::indexed::simplify_indexed_product, GiNaC::sin_eval(), GiNaC::sinh_eval(), GiNaC::spmapkey::spmapkey(), GiNaC::sqrfree_parfrac(), GiNaC::step_eval(), subs(), GiNaC::symminfo::symminfo(), GiNaC::tan_eval(), GiNaC::tanh_eval(), traverse_postorder(), traverse_preorder(), GiNaC::tryfactsubs(), GiNaC::zeta1_eval(), and GiNaC::zeta2_deriv().

8.48.3.19 operator[]() [1/4]

```
ex GiNaC::ex::operator[ ] (
            const ex & index) const  [inline]
```

References bp, and ex().

8.48.3.20 operator[]() [2/4]

```
ex GiNaC::ex::operator[] (
             size_t i) const  [inline]
```

References bp, and ex().

8.48.3.21 let_op()

```
ex & GiNaC::ex::let_op (
             size_t i)
```

Return modifiable operand/member at position i.

References bp, ex(), and makewriteable().

8.48.3.22 operator[]() [3/4]

```
ex & GiNaC::ex::operator[] (
             const ex & index)
```

References bp, ex(), and makewriteable().

8.48.3.23 operator[]() [4/4]

```
ex & GiNaC::ex::operator[] (
             size_t i)
```

References bp, ex(), and makewriteable().

8.48.3.24 lhs()

```
ex GiNaC::ex::lhs () const
```

Left hand side of relational expression.

References bp, ex(), and is_a.

Referenced by GiNaC::fsolve(), GiNaC::lhs(), GiNaC::pseries::pseries(), and GiNaC::pseries::pseries().

8.48.3.25 rhs()

```
ex GiNaC::ex::rhs () const
```

Right hand side of relational expression.

References bp, ex(), and is_a.

Referenced by GiNaC::fsolve(), GiNaC::pseries::pseries(), GiNaC::pseries::pseries(), and GiNaC::rhs().

8.48.3.26 conjugate()

```
ex GiNaC::ex::conjugate () const  [inline]
```

References bp, and ex().

Referenced by GiNaC::abs_expl_derivative(), GiNaC::abs_power(), GiNaC::acos_conjugate(), GiNaC::acosh_conjugate(), GiNaC::asin_conjugate(), GiNaC::asinh_conjugate(), GiNaC::atan_conjugate(), GiNaC::atanh_conjugate(), GiNaC::add::conjugate(), GiNaC::conjugate(), GiNaC::expair::conjugate(), GiNaC::expairseq::conjugate(), GiNaC::integral::conjugate(), GiNaC::mul::conjugate(), GiNaC::power::conjugate(), GiNaC::pseries::conjugate(), GiNaC::conjugate_eval(), GiNaC::cos_conjugate(), GiNaC::cosh_conjugate(), GiNaC::exp_conjugate(), GiNaC::lgamma_conjugate(), GiNaC::Li2_conjugate(), GiNaC::log_conjugate(), GiNaC::sin_conjugate(), GiNaC::sinh_conjugate(), GiNaC::tan_conjugate(), GiNaC::tanh_conjugate(), and GiNaC::tgamma_conjugate().

8.48.3.27 real_part()

```
ex GiNaC::ex::real_part () const  [inline]
```

References bp, and ex().

Referenced by GiNaC::abs_eval(), GiNaC::asinh_conjugate(), GiNaC::atan_conjugate(), GiNaC::conjugate_real_part(), GiNaC::mul::find_real_imag(), GiNaC::power::imag_part(), GiNaC::pseries::imag_part(), GiNaC::add::real_part(), GiNaC::power::real_part(), GiNaC::pseries::real_part(), GiNaC::real_part(), and GiNaC::real_part_eval().

8.48.3.28 imag_part()

```
ex GiNaC::ex::imag_part () const  [inline]
```

References bp, and ex().

Referenced by GiNaC::acos_conjugate(), GiNaC::acosh_conjugate(), GiNaC::asin_conjugate(), GiNaC::asinh_conjugate(), GiNaC::atan_conjugate(), GiNaC::atanh_conjugate(), GiNaC::conjugate_imag_part(), GiNaC::mul::find_real_imag(), GiNaC::add::imag_part(), GiNaC::imag_part(), GiNaC::power::imag_part(), GiNaC::imag_part_eval(), GiNaC::lgamma_conjugate(), GiNaC::Li2_conjugate(), GiNaC::log_conjugate(), and GiNaC::power::real_part().

8.48.3.29 has()

```
bool GiNaC::ex::has (
            const ex & pattern,
            unsigned options = 0) const  [inline]
```

References bp, and ex().

Referenced by GiNaC::power::degree(), GiNaC::integral::eval(), GiNaC::integral::eval_integ(), GiNaC::integral::expand(), GiNaC::Gt::has(), GiNaC::has(), GiNaC::power::is_polynomial(), and GiNaC::power::ldegree().

8.48.3.30 find()

```
bool GiNaC::ex::find (
            const ex & pattern,
            exset & found) const
```

Find all occurrences of a pattern.

The found matches are appended to the "found" list. If the expression itself matches the pattern, the children are not further examined. This function returns true when any matches were found.

References ex(), find(), match(), nops(), and op().

Referenced by GiNaC::basic::collect(), GiNaC::divide_in_z(), find(), GiNaC::find(), GiNaC::replace_with_symbol(), and GiNaC::idx::subs().

8.48.3.31 match() [1/2]

```
bool GiNaC::ex::match (
            const ex & pattern) const
```

Check whether expression matches a specified pattern.

References bp, and ex().

Referenced by find(), GiNaC::power::has(), GiNaC::match(), and GiNaC::tryfactsubs().

8.48.3.32 match() [2/2]

```
bool GiNaC::ex::match (
            const ex & pattern,
            exmap & repls) const  [inline]
```

References bp, and ex().

8.48.3.33 subs() [1/3]

```
ex GiNaC::ex::subs (
            const exmap & m,
            unsigned options = 0) const  [inline]
```

References bp, and ex().

Referenced by GiNaC::adaptivesimpson(), GiNaC::mul::algebraic_subs_mul(), GiNaC::atan_series(), GiNaC::atanh_series(), GiNaC::beta_series(), GiNaC::canonicalize_clifford(), GiNaC::clifford_to_lst(), GiNaC::collect_common_factors(), GiNaC::integral::conjugate(), GiNaC::csgn_series(), denom(), GiNaC::integral::derivative(), GiNaC::dirac_trace(), GiNaC::divide_in_z(), GiNaC::EllipticE_series(), GiNaC::EllipticK_series(), GiNaC::eta_series(), GiNaC::tensdelta::eval_indexed(), GiNaC::integral::eval_integ(), GiNaC::integral::evalf(), GiNaC::expand_dummy_sum(), GiNaC::fsolve(), GiNaC::gcd(), GiNaC::get_clifford_comp(), GiNaC::clifford::get_metric(), GiNaC::H_evalf(), GiNaC::heur_gcd_z(), GiNaC::modular_form_kernel::is_
GiNaC::user_defined_kernel::Laurent_series(), GiNaC::lgamma_series(), GiNaC::Li2_series(), GiNaC::Li_series(), GiNaC::log_series(), GiNaC::basic::normal(), normal(), GiNaC::power::normal(), numer(), numer_denom(), GiNaC::psi1_series(), GiNaC::psi2_series(), GiNaC::rename_dummy_indices(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::tensor::replace_contr_index(), GiNaC::replace_with_symbol(), GiNaC::replace_with_symbol(), GiNaC::indexed::reposition_dummy_indices, GiNaC::S_series(), GiNaC::basic::series(), GiNaC::integral::series(), GiNaC::power::series(), GiNaC::step_series(), GiNaC::basic::subs(), GiNaC::clifford::subs(), GiNaC::Gt::subs(), GiNaC::idx::subs(), GiNaC::power::subs(), GiNaC::pseries::subs(), GiNaC::relational::subs(), GiNaC::subs(), GiNaC::subs(), GiNaC::subs(), GiNaC::expairseq::subschildren(), GiNaC::subsvalue(), GiNaC::symm(), GiNaC::symmetrize_cyclic(), GiNaC::tan_series(), GiNaC::tanh_series(), GiNaC::tgamma_series(), and GiNaC::Gt::to_numeric().

8.48.3.34 subs() [2/3]

```
ex GiNaC::ex::subs (
            const lst & ls,
            const lst & lr,
            unsigned options = 0) const
```

Substitute objects in an expression (syntactic substitution) and return the result as a new expression.

References GiNaC::container< C >::begin(), bp, GiNaC::container< C >::end(), ex(), GINAC_ASSERT, is_exactly_a, GiNaC::container< C >::nops(), GiNaC::subs_options::pattern_is_not_product, and GiNaC::subs_options::pattern_is_p

8.48.3.35 subs() [3/3]

```
ex GiNaC::ex::subs (
            const ex & e,
            unsigned options = 0) const
```

Substitute objects in an expression (syntactic substitution) and return the result as a new expression.

There are two valid types of replacement arguments: 1) a relational like object==ex and 2) a list of relationals lst{object1==ex1,object2==ex2,...}.

References bp, ex(), ex_to, GINAC_ASSERT, info(), is_a, is_exactly_a, GiNaC::info_flags::list, op(), GiNaC::subs_options::pattern_is_not_product, GiNaC::subs_options::pattern_is_product, and GiNaC::info_flags::relation_equal.

8.48.3.36 map() [1/2]

```
ex GiNaC::ex::map (
            map_function & f) const  [inline]
```

References bp, and ex().

Referenced by GiNaC::canonicalize_clifford(), GiNaC::clifford_prime(), GiNaC::clifford_star_bar(), GiNaC::color_trace(), GiNaC::dirac_trace(), GiNaC::expand_dummy_sum(), GiNaC::get_clifford_comp(), and GiNaC::remove_dirac_ONE().

8.48.3.37 map() [2/2]

```
ex GiNaC::ex::map (
            ex(* f )(const ex &e)) const
```

References ex().

8.48.3.38 accept()

```
void GiNaC::ex::accept (
            visitor & v) const  [inline]
```

References bp.

Referenced by traverse_postorder(), and traverse_preorder().

8.48.3.39 traverse_preorder()

```
void GiNaC::ex::traverse_preorder (
            visitor & v) const
```

Traverse expression tree with given visitor, preorder traversal.

References accept(), nops(), op(), and traverse_preorder().

Referenced by traverse(), and traverse_preorder().

8.48.3.40 traverse_postorder()

```
void GiNaC::ex::traverse_postorder (
            visitor & v) const
```

Traverse expression tree with given visitor, postorder traversal.

References accept(), nops(), op(), and traverse_postorder().

Referenced by traverse_postorder().

8.48.3.41 traverse()

```
void GiNaC::ex::traverse (
            visitor & v) const  [inline]
```

References traverse_preorder().

8.48.3.42 is_polynomial()

```
bool GiNaC::ex::is_polynomial (
            const ex & vars) const
```

Check whether expression is a polynomial.

References bp, ex(), ex_to, and is_a.

Referenced by GiNaC::is_polynomial(), and GiNaC::power::is_polynomial().

8.48.3.43 degree()

```
int GiNaC::ex::degree (
            const ex & s) const  [inline]
```

References bp, and ex().

Referenced by GiNaC::basic::collect(), content(), GiNaC::degree(), GiNaC::integral::degree(), GiNaC::mul::degree(), GiNaC::power::degree(), GiNaC::divide(), GiNaC::divide_in_z(), GiNaC::get_symbol_stats(), GiNaC::heur_gcd_z(), lcoeff(), GiNaC::prem(), GiNaC::quo(), GiNaC::rem(), GiNaC::resultant(), GiNaC::sprem(), and GiNaC::sr_gcd().

8.48.3.44 ldegree()

```
int GiNaC::ex::ldegree (
            const ex & s) const  [inline]
```

References bp, and ex().

Referenced by content(), GiNaC::gcd_pf_pow(), GiNaC::get_symbol_stats(), GiNaC::ldegree(), GiNaC::mul::ldegree(), GiNaC::power::ldegree(), GiNaC::Order_series(), GiNaC::resultant(), GiNaC::mul::series(), GiNaC::power::series(), and tcoeff().

8.48.3.45 coeff()

```
ex GiNaC::ex::coeff (
            const ex & s,
            int n = 1) const  [inline]
```

References bp, and ex().

Referenced by GiNaC::Bernoulli_polynomial(), GiNaC::add::coeff(), GiNaC::coeff(), GiNaC::mul::coeff(), GiNaC::ncmul::coeff(), GiNaC::pseries::coeff(), GiNaC::basic::collect(), content(), GiNaC::pseries::convert_to_poly(), GiNaC::mul::derivative(), GiNaC::pseries::derivative(), GiNaC::divide(), GiNaC::divide_in_z(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::pseries::eval_integ(), GiNaC::pseries::evalm(), GiNaC::pseries::expand(), GiNaC::expairseq::expandch GiNaC::generalised_Bernoulli_number(), GiNaC::add::imag_part(), lcoeff(), GiNaC::lsolve(), GiNaC::expairseq::make_flat(), GiNaC::pseries::normal(), GiNaC::prem(), GiNaC::pseries::print_series(), GiNaC::quo(), GiNaC::add::real_part(), GiNaC::rem(), GiNaC::resultant(), GiNaC::mul::series(), GiNaC::integration_kernel::series_coeff(), GiNaC::indexed::simplify_indexed, GiNaC::sprem(), GiNaC::sqrfree_parfrac(), GiNaC::expairseq::subschildren(), and tcoeff().

8.48.3.46 lcoeff()

```
ex GiNaC::ex::lcoeff (
            const ex & s) const  [inline]
```

References coeff(), degree(), and ex().

Referenced by content(), GiNaC::get_symbol_stats(), GiNaC::sr_gcd(), and unit().

8.48.3.47 tcoeff()

```
ex GiNaC::ex::tcoeff (
            const ex & s) const  [inline]
```

References coeff(), ex(), and ldegree().

8.48.3.48 expand()

```
ex GiNaC::ex::expand (
            unsigned options = 0) const
```

Expand an expression.

Parameters

options	see GiNaC::expand_options

References bp, ex(), and GiNaC::status_flags::expanded.

Referenced by GiNaC::abs_expand(), GiNaC::binomial_sym(), GiNaC::canonicalize_clifford(), GiNaC::clifford_star_bar(), GiNaC::color_trace(), content(), GiNaC::matrix::determinant_minor(), GiNaC::divide(), GiNaC::divide_in_z(), GiNaC::exp_expand(), GiNaC::expand(), GiNaC::indexed::expand(), GiNaC::integral::expand(), GiNaC::mul::expand(), GiNaC::power::expand(), GiNaC::expand_dummy_sum(), GiNaC::mul::expandchildren(), GiNaC::ncmul::expandchildren(), GiNaC::mul::find_real_imag(), GiNaC::frac_cancel(), GiNaC::gcd(), GiNaC::get_clifford_comp(), GiNaC::heur_gcd_z(), GiNaC::log_expand(), GiNaC::lsolve(), GiNaC::expand_map_function::operator()(), GiNaC::prem(), GiNaC::quo(), GiNaC::rem(), GiNaC::resultant(), GiNaC::basic::series(), GiNaC::mul::series(), GiNaC::power::series(), GiNaC::indexed::simplify_ind GiNaC::sprem(), GiNaC::sqrfree_parfrac(), GiNaC::sr_gcd(), GiNaC::matrix::trace(), unit(), and unitcontprim().

8.48.3.49 collect()

```
ex GiNaC::ex::collect (
            const ex & s,
            bool distributed = false) const  [inline]
```

References bp, and ex().

Referenced by GiNaC::matrix::charpoly(), GiNaC::basic::collect(), and GiNaC::collect().

8.48.3.50 diff()

```
ex GiNaC::ex::diff (
            const symbol & s,
            unsigned nth = 1) const
```

Compute partial derivative of an expression.

Parameters

s	symbol by which the expression is derived
nth	order of derivative (default 1)

Returns

partial derivative as a new expression

References bp, and ex().

Referenced by GiNaC::abs_expl_derivative(), GiNaC::conjugate_expl_derivative(), GiNaC::integral::derivative(), GiNaC::power::derivative(), GiNaC::basic::diff(), GiNaC::diff(), GiNaC::fsolve(), GiNaC::imag_part_expl_derivative(), GiNaC::log_series(), GiNaC::Order_expl_derivative(), GiNaC::real_part_expl_derivative(), GiNaC::basic::series(), and GiNaC::sqrfree_yun().

8.48.3.51 series()

```
ex GiNaC::ex::series (
              const ex & r,
              int order,
              unsigned options = 0) const
```

Compute the truncated series expansion of an expression.

This function returns an expression containing an object of class pseries to represent the series. If the series does not terminate within the given truncation order, the last term of the series will be an order term.

Parameters

r	expansion relation, lhs holds variable and rhs holds point
order	truncation order of series calculations
options	of class series_options

Returns

an expression holding a pseries object

References GiNaC::_ex0, bp, ex(), ex_to, is_a, and GiNaC::relational::relational().

Referenced by GiNaC::Bernoulli_polynomial(), GiNaC::EllipticE_series(), GiNaC::EllipticK_series(), GiNaC::generalised_Bernoulli_nu GiNaC::Eisenstein_h_kernel::Laurent_series(), GiNaC::Eisenstein_kernel::Laurent_series(), GiNaC::integration_kernel::Laurent_seri GiNaC::modular_form_kernel::Laurent_series(), GiNaC::user_defined_kernel::Laurent_series(), GiNaC::Li2_series(), GiNaC::Li_series(), GiNaC::log_series(), GiNaC::psi1_series(), GiNaC::Eisenstein_h_kernel::q_expansion_modular_form(), GiNaC::S_series(), GiNaC::add::series(), GiNaC::Eisenstein_h_kernel::series(), GiNaC::Eisenstein_kernel::series(), GiNaC::integral::series(), GiNaC::modular_form_kernel::series(), GiNaC::mul::series(), GiNaC::power::series(), GiNaC::pseries::series(), GiNaC::series(), and GiNaC::tgamma_series().

8.48.3.52 normal()

```
ex GiNaC::ex::normal () const
```

Normalization of rational functions.

This function converts an expression to its normal form "numerator/denominator", where numerator and denominator are (relatively prime) polynomials. Any subexpressions which are not rational functions (like non-rational numbers, non-integer powers or functions like sin(), cos() etc.) are replaced by temporary symbols which are resubstituted by the (normalized) subexpressions before normal() returns (this way, any expression can be treated as a rational function). normal() is applied recursively to arguments of functions etc.

Returns

normalized expression

References bp, ex(), GINAC_ASSERT, is_a, GiNaC::subs_options::no_pattern, GiNaC::container< C >::nops(), GiNaC::container< C >::op(), op(), and subs().

Referenced by GiNaC::matrix::determinant(), GiNaC::add::normal(), GiNaC::mul::normal(), GiNaC::normal(), GiNaC::matrix::trace(), and GiNaC::Gt::zisToFundamental().

8.48.3.53 to_rational()

```
ex GiNaC::ex::to_rational (
            exmap & repl) const
```

Rationalization of non-rational functions.

This function converts a general expression to a rational function by replacing all non-rational subexpressions (like non-rational numbers, non-integer powers or functions like sin(), cos() etc.) to temporary symbols. This makes it possible to use functions like gcd() and divide() on non-rational functions by applying to_rational() on the arguments, calling the desired function and re-substituting the temporary symbols in the result. To make the last step possible, all temporary symbols and their associated expressions are collected in the map specified by the repl parameter, ready to be passed as an argument to ex::subs().

Parameters

repl	collects all temporary symbols and their replacements

Returns

rationalized expression

References bp, and ex().

Referenced by GiNaC::canonicalize_clifford(), GiNaC::expairseq::to_rational(), GiNaC::power::to_rational(), and GiNaC::to_rational().

8.48.3.54 to_polynomial()

```
ex GiNaC::ex::to_polynomial (
            exmap & repl) const
```

References bp, and ex().

Referenced by GiNaC::find_common_factor(), GiNaC::expairseq::to_polynomial(), GiNaC::power::to_polynomial(), and GiNaC::to_polynomial().

8.48.3.55 numer()

```
ex GiNaC::ex::numer () const
```

Get numerator of an expression.

If the expression is not of the normal form "numerator/denominator", it is first converted to this form and then the numerator is returned.

See also

ex::normal

Returns

numerator

References bp, ex(), GINAC_ASSERT, is_a, GiNaC::subs_options::no_pattern, GiNaC::container< C >::nops(), GiNaC::container< C >::op(), op(), and subs().

Referenced by GiNaC::numer().

8.48.3.56 denom()

```
ex GiNaC::ex::denom () const
```

Get denominator of an expression.

If the expression is not of the normal form "numerator/denominator", it is first converted to this form and then the denominator is returned.

See also

ex::normal

Returns

denominator

References bp, ex(), GINAC_ASSERT, is_a, GiNaC::subs_options::no_pattern, GiNaC::container< C >::nops(), GiNaC::container< C >::op(), op(), and subs().

Referenced by GiNaC::denom().

8.48.3.57 numer_denom()

```
ex GiNaC::ex::numer_denom () const
```

Get numerator and denominator of an expression.

If the expression is not of the normal form "numerator/denominator", it is first converted to this form and then a list [numerator, denominator] is returned.

See also

ex::normal

Returns

a list [numerator, denominator]

References bp, ex(), GINAC_ASSERT, is_a, GiNaC::subs_options::no_pattern, GiNaC::container< C >::nops(), GiNaC::container< C >::op(), and subs().

Referenced by GiNaC::numer_denom().

8.48.3.58 unit()

```
ex GiNaC::ex::unit (
            const ex & x) const
```

Compute unit part (= sign of leading coefficient) of a multivariate polynomial in Q[x].

The product of unit part, content part, and primitive part is the polynomial itself.

Parameters

x	main variable

Returns

unit part

See also

ex::content, ex::primpart, ex::unitcontprim

References GiNaC::_ex1, GiNaC::_ex_1, ex(), expand(), GiNaC::get_first_symbol(), info(), is_exactly_a, lcoeff(), GiNaC::info_flags::negative, and unit().

Referenced by content(), GiNaC::frac_cancel(), primpart(), unit(), and unitcontprim().

8.48.3.59 content()

```
ex GiNaC::ex::content (
            const ex & x) const
```

Compute content part (= unit normal GCD of all coefficients) of a multivariate polynomial in Q[x].

The product of unit part, content part, and primitive part is the polynomial itself.

Parameters

x	main variable

Returns

content part

See also

ex::unit, ex::primpart, ex::unitcontprim

References GiNaC::_ex0, coeff(), degree(), ex(), expand(), GiNaC::gcd(), info(), GiNaC::info_flags::integer, integer_content(), is_exactly_a, is_zero(), lcoeff(), ldegree(), GiNaC::info_flags::negative, and unit().

Referenced by GiNaC::sr_gcd(), and unitcontprim().

8.48.3.60 integer_content()

```
numeric GiNaC::ex::integer_content () const
```

Compute the integer content (= GCD of all numeric coefficients) of an expanded polynomial.

For a polynomial with rational coefficients, this returns g/l where g is the GCD of the coefficients' numerators and l is the LCM of the coefficients' denominators.

Returns

integer content

References bp.

Referenced by content(), GiNaC::power::eval(), GiNaC::gcd(), and GiNaC::heur_gcd_z().

8.48.3.61 primpart() [1/2]

```
ex GiNaC::ex::primpart (
             const ex & x) const
```

Compute primitive part of a multivariate polynomial in Q[x].

The result will be a unit-normal polynomial with a content part of 1. The product of unit part, content part, and primitive part is the polynomial itself.

Parameters

x	main variable

Returns

primitive part

See also

ex::unit, ex::content, ex::unitcontprim

References ex(), and unitcontprim().

Referenced by GiNaC::sr_gcd().

8.48.3.62 primpart() [2/2]

```
ex GiNaC::ex::primpart (
              const ex & x,
              const ex & c) const
```

Compute primitive part of a multivariate polynomial in Q[x] when the content part is already known.

This function is faster in computing the primitive part than the previous function.

Parameters

x	main variable
c	previously computed content part

Returns

primitive part

References GiNaC::_ex0, GiNaC::_ex1, ex(), is_exactly_a, is_zero(), GiNaC::quo(), and unit().

8.48.3.63 unitcontprim()

```
void GiNaC::ex::unitcontprim (
              const ex & x,
              ex & u,
              ex & c,
              ex & p) const
```

Compute unit part, content part, and primitive part of a multivariate polynomial in Q[x].

The product of the three parts is the polynomial itself.

Parameters

x	main variable
u	unit part (returned)
c	content part (returned)
p	primitive part (returned)

See also

ex::unit, ex::content, ex::primpart

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::_ex_1, GiNaC::abs(), content(), ex(), ex_to, expand(), info(), is_exactly_a, is_zero(), GiNaC::info_flags::negative, GiNaC::quo(), and unit().

Referenced by GiNaC::gcd(), and primpart().

8.48.3.64 smod()

```
ex GiNaC::ex::smod (
            const numeric & xi) const  [inline]
```

References bp, and ex().

Referenced by GiNaC::interpolate().

8.48.3.65 max_coefficient()

```
numeric GiNaC::ex::max_coefficient () const
```

Return maximum (absolute value) coefficient of a polynomial.

This function is used internally by heur_gcd().

Returns

maximum coefficient

See also

heur_gcd

References bp.

Referenced by GiNaC::heur_gcd_z().

8.48.3.66 get_free_indices()

```
exvector GiNaC::ex::get_free_indices () const  [inline]
```

References bp.

Referenced by antisymmetrize(), GiNaC::clifford_unit(), GiNaC::get_all_dummy_indices_safely(), GiNaC::add::get_free_indices(), GiNaC::integral::get_free_indices(), GiNaC::mul::get_free_indices(), GiNaC::ncmul::get_free_indices(), GiNaC::clifford::get_metric(), GiNaC::make_flat_inserter::handle_factor(), GiNaC::clifford::same_metric(), symmetrize(), and symmetrize_cyclic().

8.48.3.67 simplify_indexed() [1/2]

```
ex GiNaC::ex::simplify_indexed (
            unsigned options = 0) const
```

Simplify/canonicalize expression containing indexed objects.

This performs contraction of dummy indices where possible and checks whether the free indices in sums are consistent.

Parameters

options	Simplification options (currently unused)

Returns

simplified expression

References ex(), and GiNaC::simplify_indexed().

Referenced by GiNaC::canonicalize_clifford(), GiNaC::tensepsilon::contract_with(), GiNaC::simplify_indexed(), and GiNaC::simplify_indexed().

8.48.3.68 simplify_indexed() [2/2]

```
ex GiNaC::ex::simplify_indexed (
            const scalar_products & sp,
            unsigned options = 0) const
```

Simplify/canonicalize expression containing indexed objects.

This performs contraction of dummy indices where possible, checks whether the free indices in sums are consistent, and automatically replaces scalar products by known values if desired.

Parameters

sp	Scalar products to be replaced automatically
options	Simplification options (currently unused)

Returns

simplified expression

References ex(), and GiNaC::simplify_indexed().

8.48.3.69 compare()

```
int GiNaC::ex::compare (
            const ex & other) const  [inline]
```

References bp, ex(), and share().

Referenced by GiNaC::expair::compare(), GiNaC::expairseq::construct_from_2_ex(), GiNaC::expair::is_less(), GiNaC::error_and_integral_is_less::operator()(), GiNaC::ex_is_less::operator()(), GiNaC::expair_rest_is_less::operator()(), GiNaC::symminfo_is_less_by_orig::operator()(), GiNaC::symminfo_is_less_by_symmterm::operator()(), GiNaC::terminfo_is_less::ope GiNaC::Gt_detail::pathintegral_term::operator<, GiNaC::spmapkey::operator<(), and GiNaC::spmapkey::spmapkey().

8.48.3.70 is_equal()

```
bool GiNaC::ex::is_equal (
            const ex & other) const  [inline]
```

References bp, ex(), and share().

Referenced by GiNaC::abs_power(), GiNaC::acos_eval(), GiNaC::acosh_eval(), GiNaC::matrix::add_indexed(), GiNaC::asin_eval(), GiNaC::atan2_eval(), GiNaC::atan_eval(), GiNaC::atan_series(), GiNaC::atanh_eval(), GiNaC::atanh_series(), GiNaC::beta_eval(), GiNaC::power::coeff(), GiNaC::pseries::coeff(), GiNaC::add::combine_ex_with_coeff_to_, GiNaC::mul::combine_ex_with_coeff_to_pair(), GiNaC::mul::combine_pair_with_coeff_to_pair(), GiNaC::mul::conjugate(), GiNaC::diracgamma::contract_with(), GiNaC::power::degree(), GiNaC::pseries::degree(), GiNaC::divide(), GiNaC::divide_in_z(), GiNaC::mul::do_print_csrc(), GiNaC::power::do_print_csrc(), GiNaC::power::do_print_csrc_cl_N(), GiNaC::power::do_print_dflt(), GiNaC::power::do_print_latex(), GiNaC::expairseq::do_print_tree(), GiNaC::epsilon_tensor(), GiNaC::epsilon_tensor(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::matrix::eval_indexed(), GiNaC::tensdelta::eval_indexed(), GiNaC::tensmetric::eval_indexed(), GiNaC::clifford::eval_ncmul(), GiNaC::mul::expand(), GiNaC::power::expand_add_2(), GiNaC::find_common_factor(), GiNaC::frac_cancel(), GiNaC::gcd(), GiNaC::gcd_pf_pow(), GiNaC::gcd_pf_pow_pow(), GiNaC::make_flat_inserter::handle_factor(), GiNaC::power::imag_part(), GiNaC::mul::info(), GiNaC::expair::is_canonical_numeric(), GiNaC::pseries::is_compatible_to(), GiNaC::idx::is_dummy_pair_same_type(), GiNaC::expair::is_equal(), GiNaC::expairseq::is_equa is_zero(), GiNaC::power::ldegree(), GiNaC::pseries::ldegree(), GiNaC::Li2_eval(), GiNaC::Li2_series(), GiNaC::Li_eval(), GiNaC::log_eval(), GiNaC::lorentz_eps(), GiNaC::expairseq::map(), GiNaC::expairseq::match(), GiNaC::idx::match_same_type(), GiNaC::minimal_dim(), GiNaC::pseries::mul_series(), GiNaC::multiply_lcm(), GiNaC::expairseq::nops(), GiNaC::power::normal(), GiNaC::expairseq::op(), GiNaC::ex_is_equal::operator()(), GiNaC::idx_is_equal_ignore_dim::operator()(), GiNaC::op0_is_equal::oper GiNaC::spmapkey::operator==(), GiNaC::add::print_add(), GiNaC::mul::print_overall_coeff(), GiNaC::expairseq::printseq(), GiNaC::product_to_exvector(), GiNaC::psi2_eval(), GiNaC::quo(), GiNaC::power::real_part(), GiNaC::mul::recombine_pair_to_ex(), GiNaC::rem(), GiNaC::replace_with_symbol(), GiNaC::clifford::same_metric(), GiNaC::power::series(), GiNaC::pseries::series(), GiNaC::indexed::simplify_indexed, GiNaC::add::split_ex_to_pair(), and GiNaC::sqrfree_yun().

8.48.3.71 is_zero()

```
bool GiNaC::ex::is_zero () const  [inline]
```

References GiNaC::_ex0, ex(), and is_equal().

Referenced by GiNaC::acos_conjugate(), GiNaC::acos_eval(), GiNaC::acosh_conjugate(), GiNaC::acosh_eval(), GiNaC::pseries::add_series(), GiNaC::asin_conjugate(), GiNaC::asin_eval(), GiNaC::asinh_eval(), GiNaC::atan2_eval(), GiNaC::atan_eval(), GiNaC::atanh_conjugate(), GiNaC::atanh_eval(), GiNaC::clifford_inverse(), GiNaC::clifford_to_lst(), GiNaC::add::coeff(), GiNaC::mul::coeff(), GiNaC::ncmul::coeff(), content(), GiNaC::cosh_eval(), GiNaC::add::degree(), GiNaC::fderivative::derivative(), GiNaC::function::derivative(), GiNaC::pseries::derivative(), GiNaC::matrix::determinant(), GiNaC::matrix::determinant_minor(), GiNaC::basic::diff(), GiNaC::divide(), GiNaC::divide_in_z(), GiNaC::add::do_print_csrc(), GiNaC::add::eval(), GiNaC::function::eval(), GiNaC::indexed::eval(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::pseries::evalm(), GiNaC::Gt_detail::pathintegral_term::evaluate(), GiNaC::exp_eval(), GiNaC::mul::expand(), GiNaC::pseries::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), GiNaC::find_common_factor(), GiNaC::mul::find_real_imag(), GiNaC::frac_cancel(), GiNaC::gcd(), GiNaC::get_clifford_comp(), GiNaC::heur_gcd_z(), GiNaC::add::imag_part(), GiNaC::add::info(), GiNaC::Gt_detail::pathintegral_term::integrate(), GiNaC::interpolate(), GiNaC::is_zero(), is_zero_matrix(), GiNaC::add::ldegree(), GiNaC::lgamma_conjugate(), GiNaC::Li2_conjugate(), GiNaC::Li2_eval(), GiNaC::Li2_series(), GiNaC::Li3_eval(), GiNaC::Li_series(), GiNaC::log_conjugate(), GiNaC::log_eval(), GiNaC::log_series(), GiNaC::matrix::markowitz_elimination(), GiNaC::pseries::mul_series(), GiNaC::pseries::normal(), GiNaC::relational::operator safe_bool(), GiNaC::Gt_detail::pathintegral_term::operator<, GiNaC::Gt_detail::pathintegral_term::operator<<, GiNaC::Order_eval(), GiNaC::Gt_detail::pathintegral_term::pathintegral_term(), GiNaC::prem(), primpart(), GiNaC::add::print_add(), GiNaC::pseries::print_series(), GiNaC::psi2_eval(), GiNaC::quo(), GiNaC::add::real_part(), GiNaC::rem(), GiNaC::S_eval(), GiNaC::S_series(), GiNaC::clifford::same_metric(), GiNaC::basic::series(), GiNaC::power::series(), GiNaC::symbol::series(), GiNaC::indexed::simplify_indexed, GiNaC::indexed::simplify_indexed_product, GiNaC::sinh_eval(), GiNaC::add::smod(), GiNaC::sprem(), GiNaC::sqrfree_yun(), GiNaC::sr_gcd(), GiNaC::tanh_eval(), GiNaC::Gt::tauToFundamental(), unitcontprim(), GiNaC::indexed::validate(), and GiNaC::zetaderiv_eval().

8.48.3.72 is_zero_matrix()

```
bool GiNaC::ex::is_zero_matrix () const
```

Check whether expression is zero or zero matrix.

References evalm(), ex(), ex_to, is_a, and is_zero().

8.48.3.73 symmetrize() [1/2]

```
ex GiNaC::ex::symmetrize () const
```

Symmetrize expression over its free indices.

References ex(), get_free_indices(), and GiNaC::symmetrize().

Referenced by GiNaC::symmetrize(), and GiNaC::symmetrize().

8.48.3.74 symmetrize() [2/2]

```
ex GiNaC::ex::symmetrize (
             const lst & l) const
```

Symmetrize expression over a list of objects (symbols, indices).

References GiNaC::container< C >::begin(), GiNaC::container< C >::end(), ex(), and GiNaC::symm().

8.48.3.75 antisymmetrize() [1/2]

```
ex GiNaC::ex::antisymmetrize () const
```

Antisymmetrize expression over its free indices.

References GiNaC::antisymmetrize(), ex(), and get_free_indices().

Referenced by GiNaC::antisymmetrize(), and GiNaC::antisymmetrize().

8.48.3.76 antisymmetrize() [2/2]

```
ex GiNaC::ex::antisymmetrize (
             const lst & l) const
```

Antisymmetrize expression over a list of objects (symbols, indices).

References GiNaC::container< C >::begin(), GiNaC::container< C >::end(), ex(), and GiNaC::symm().

8.48.3.77 symmetrize_cyclic() [1/2]

```
ex GiNaC::ex::symmetrize_cyclic () const
```

Symmetrize expression by cyclic permutation over its free indices.

References ex(), get_free_indices(), and GiNaC::symmetrize_cyclic().

Referenced by GiNaC::symmetrize_cyclic(), and GiNaC::symmetrize_cyclic().

8.48.3.78 symmetrize_cyclic() [2/2]

```
ex GiNaC::ex::symmetrize_cyclic (
            const lst & l) const
```

Symmetrize expression by cyclic permutation over a list of objects (symbols, indices).

References GiNaC::container< C >::begin(), GiNaC::container< C >::end(), ex(), and GiNaC::symmetrize_cyclic().

8.48.3.79 return_type()

```
unsigned GiNaC::ex::return_type () const  [inline]
```

References bp.

Referenced by GiNaC::ncmul::append_factors(), GiNaC::canonicalize_clifford(), GiNaC::ncmul::count_factors(), GiNaC::power::eval(), GiNaC::exmul(), GiNaC::matrix::mul_scalar(), GiNaC::indexed::return_type(), GiNaC::integral::return_type(), GiNaC::power::return_type(), and GiNaC::relational::return_type().

8.48.3.80 return_type_tinfo()

```
return_type_t GiNaC::ex::return_type_tinfo () const  [inline]
```

References bp.

Referenced by GiNaC::canonicalize_clifford(), GiNaC::color_trace(), GiNaC::dirac_trace(), GiNaC::indexed::return_type_tinfo(), GiNaC::integral::return_type_tinfo(), GiNaC::power::return_type_tinfo(), and GiNaC::relational::return_type_tinfo().

8.48.3.81 gethash()

```
unsigned GiNaC::ex::gethash () const  [inline]
```

References bp.

Referenced by GiNaC::basic::calchash(), GiNaC::expairseq::calchash(), GiNaC::function::calchash(), GiNaC::Gt::calchash(), GiNaC::idx::calchash(), and GiNaC::relational::calchash().

8.48.3.82 construct_from_basic()

```
ptr< basic > GiNaC::ex::construct_from_basic (
            const basic & other)  [static], [private]
```

Helper function for the ex-from-basic constructor.

This is where GiNaC's automatic evaluator and memory management are implemented.

See also

ex::ex(const basic &)

References bp, GiNaC::basic::duplicate(), GiNaC::status_flags::dynallocated, GiNaC::basic::eval(), GiNaC::status_flags::evaluated, ex(), GiNaC::basic::flags, GiNaC::refcounted::get_refcount(), GINAC_ASSERT, GiNaC::ptr< T >::ptr(), and GiNaC::basic::setflag().

Referenced by ex().

8.48.3.83 construct_from_int()

```
basic & GiNaC::ex::construct_from_int (
            int i)  [static], [private]
```

References GiNaC::_num0_p, GiNaC::_num10_p, GiNaC::_num11_p, GiNaC::_num12_p, GiNaC::_num1_p, GiNaC::_num2_p, GiNaC::_num3_p, GiNaC::_num4_p, GiNaC::_num5_p, GiNaC::_num6_p, GiNaC::_num7_p, GiNaC::_num8_p, GiNaC::_num9_p, GiNaC::_num_10_p, GiNaC::_num_11_p, GiNaC::_num_12_p, GiNaC::_num_1_p, GiNaC::_num_2_p, GiNaC::_num_3_p, GiNaC::_num_4_p, GiNaC::_num_5_p, GiNaC::_num_6_p, GiNaC::_num_7_p, GiNaC::_num_8_p, GiNaC::_num_9_p, and GiNaC::dynallocate().

Referenced by construct_from_longlong(), and ex().

8.48.3.84 construct_from_uint()

```
basic & GiNaC::ex::construct_from_uint (
            unsigned int i)  [static], [private]
```

References GiNaC::_num0_p, GiNaC::_num10_p, GiNaC::_num11_p, GiNaC::_num12_p, GiNaC::_num1_p, GiNaC::_num2_p, GiNaC::_num3_p, GiNaC::_num4_p, GiNaC::_num5_p, GiNaC::_num6_p, GiNaC::_num7_p, GiNaC::_num8_p, GiNaC::_num9_p, and GiNaC::dynallocate().

Referenced by construct_from_ulonglong(), and ex().

8.48.3.85 construct_from_long()

```
basic & GiNaC::ex::construct_from_long (
            long i)  [static], [private]
```

References GiNaC::_num0_p, GiNaC::_num10_p, GiNaC::_num11_p, GiNaC::_num12_p, GiNaC::_num1_p, GiNaC::_num2_p, GiNaC::_num3_p, GiNaC::_num4_p, GiNaC::_num5_p, GiNaC::_num6_p, GiNaC::_num7_p, GiNaC::_num8_p, GiNaC::_num9_p, GiNaC::_num_10_p, GiNaC::_num_11_p, GiNaC::_num_12_p, GiNaC::_num_1_p, GiNaC::_num_2_p, GiNaC::_num_3_p, GiNaC::_num_4_p, GiNaC::_num_5_p, GiNaC::_num_6_p, GiNaC::_num_7_p, GiNaC::_num_8_p, GiNaC::_num_9_p, and GiNaC::dynallocate().

Referenced by ex().

8.48.3.86 construct_from_ulong()

```
basic & GiNaC::ex::construct_from_ulong (
            unsigned long i)  [static], [private]
```

References GiNaC::_num0_p, GiNaC::_num10_p, GiNaC::_num11_p, GiNaC::_num12_p, GiNaC::_num1_p, GiNaC::_num2_p, GiNaC::_num3_p, GiNaC::_num4_p, GiNaC::_num5_p, GiNaC::_num6_p, GiNaC::_num7_p, GiNaC::_num8_p, GiNaC::_num9_p, and GiNaC::dynallocate().

Referenced by ex().

8.48.3.87 construct_from_longlong()

```
basic & GiNaC::ex::construct_from_longlong (
            long long i)  [static], [private]
```

References construct_from_int(), and GiNaC::dynallocate().

Referenced by ex().

8.48.3.88 construct_from_ulonglong()

```
basic & GiNaC::ex::construct_from_ulonglong (
            unsigned long long i)  [static], [private]
```

References construct_from_uint(), and GiNaC::dynallocate().

Referenced by ex().

8.48.3.89 construct_from_double()

```
basic & GiNaC::ex::construct_from_double (
            double d)  [static], [private]
```

References GiNaC::dynallocate().

Referenced by ex().

8.48.3.90 construct_from_string_and_lst()

```
ptr< basic > GiNaC::ex::construct_from_string_and_lst (
            const std::string & s,
            const ex & l)  [static], [private]
```

References ex(), makewriteable(), and share().

Referenced by ex().

8.48.3.91 makewriteable()

```
void GiNaC::ex::makewriteable ()  [private]
```

Make this ex writable (if more than one ex handle the same basic) by unlinking the object and creating an unshared copy of it.

References bp, GiNaC::status_flags::dynallocated, and GINAC_ASSERT.

Referenced by construct_from_string_and_lst(), let_op(), operator[ ](), and operator[ ]().

8.48.3.92 share()

```
void GiNaC::ex::share (
            const ex & other) const  [private]
```

Share equal objects between expressions.

See also

ex::compare(const ex &)

References bp, ex(), and GiNaC::status_flags::not_shareable.

Referenced by compare(), construct_from_string_and_lst(), and is_equal().

8.48.4 Friends And Related Symbol Documentation

8.48.4.1 archive_node

```
friend class archive_node  [friend]
```

References archive_node, are_ex_trivially_equal, ex(), ex_to, is_a, and is_exactly_a.

Referenced by archive_node.

8.48.4.2 are_ex_trivially_equal

```
bool are_ex_trivially_equal (
            const ex & e1,
            const ex & e2)  [friend]
```

Compare two objects of class quickly without doing a deep tree traversal.

Returns

"true" if they are equal "false" if equality cannot be established quickly (e1 and e2 may still be equal, in this case.

References are_ex_trivially_equal, bp, and ex().

Referenced by archive_node, and are_ex_trivially_equal.

8.48.4.3 ex_to

```
template<class T>
const T & ex_to (
            const ex & e)  [friend]
```

Return a reference to the basic-derived class T object embedded in an expression.

This is fast but unsafe: the result is undefined if the expression does not contain a T object at its top level. Hence, you should generally check the type of e first. Also, you shouldn't cache the returned reference because GiNaC's garbage collector may destroy the referenced object any time it's used in another expression.

Parameters

e	expression

Returns

reference to object of class T

See also

is_exactly_a<class T>()

References bp, ex(), ex_to, GINAC_ASSERT, and is_a.

Referenced by archive_node, ex_to, is_polynomial(), is_zero_matrix(), series(), subs(), and unitcontprim().

8.48.4.4 is_a

```
template<class T>
bool is_a (
            const ex & obj)  [friend]
```

Check if ex is a handle to a T, including base classes.

References bp, ex(), and is_a.

Referenced by archive_node, denom(), ex_to, is_a, is_polynomial(), is_zero_matrix(), lhs(), normal(), numer(), numer_denom(), rhs(), series(), and subs().

8.48.4.5 is_exactly_a

```
template<class T>
bool is_exactly_a (
            const ex & obj)  [friend]
```

Check if ex is a handle to a T, not including base classes.

References bp, ex(), and is_exactly_a.

Referenced by archive_node, content(), is_exactly_a, primpart(), subs(), subs(), unit(), and unitcontprim().

8.48.5 Member Data Documentation

8.48.5.1 bp

```
ptr<basic> GiNaC::ex::bp  [mutable], [private]
```

pointer to basic object managed by this

Referenced by accept(), GiNaC::archive_node::archive_node(), are_ex_trivially_equal, coeff(), collect(), compare(), conjugate(), construct_from_basic(), dbgprint(), dbgprinttree(), degree(), denom(), diff(), eval(), eval_integ(), eval_ncmul(), evalf(), evalm(), ex(), ex(), ex(), ex(), ex(), ex(), ex(), ex(), ex(), ex_to, expand(), get_free_indices(), gethash(), has(), GiNaC::archive_node::has_same_ex_as(), imag_part(), info(), integer_content(), is_a, is_equal(), is_exactly_a, is_polynomial(), ldegree(), let_op(), lhs(), makewriteable(), map(), match(), match(), max_coefficient(), nops(), normal(), numer(), numer_denom(), op(), operator[ ](), operator[ ](), operator[ ](), operator[ ](), print(), GiNaC::archive_node::printraw(), real_part(), return_type(), return_type_tinfo(), rhs(), series(), share(), smod(), subs(), subs(), subs(), swap(), to_polynomial(), and to_rational().

The documentation for this class was generated from the following files:

- ex.h
- ex.cpp
- indexed.cpp
- normal.cpp
- pseries.cpp
- symmetry.cpp

8.49 GiNaC::ex_base_is_less Struct Reference

Public Member Functions

- bool operator() (const ex &lh, const ex &rh) const

8.49.1 Member Function Documentation

8.49.1.1 operator()()

```
bool GiNaC::ex_base_is_less::operator() (
            const ex & lh,
            const ex & rh) const  [inline]
```

References GiNaC::is_a(), and GiNaC::ex::op().

The documentation for this struct was generated from the following file:

- indexed.cpp

8.50 GiNaC::ex_is_equal Struct Reference

```
#include <ex.h>
```

Public Member Functions

- bool operator() (const ex &lh, const ex &rh) const

8.50.1 Member Function Documentation

8.50.1.1 operator()()

```
bool GiNaC::ex_is_equal::operator() (
            const ex & lh,
            const ex & rh) const  [inline]
```

References GiNaC::ex::is_equal().

The documentation for this struct was generated from the following file:

- ex.h

8.51 GiNaC::ex_is_less Struct Reference

```
#include <ex.h>
```

Public Member Functions

- bool operator() (const ex &lh, const ex &rh) const

8.51.1 Member Function Documentation

8.51.1.1 operator()()

```
bool GiNaC::ex_is_less::operator() (
            const ex & lh,
            const ex & rh) const  [inline]
```

References GiNaC::ex::compare().

The documentation for this struct was generated from the following file:

- ex.h

8.52 GiNaC::ex_swap Struct Reference

```
#include <ex.h>
```

Public Member Functions

- void operator() (ex &lh, ex &rh) const

8.52.1 Member Function Documentation

8.52.1.1 operator()()

```
void GiNaC::ex_swap::operator() (
             ex & lh,
             ex & rh) const  [inline]
```

References GiNaC::ex::swap().

The documentation for this struct was generated from the following file:

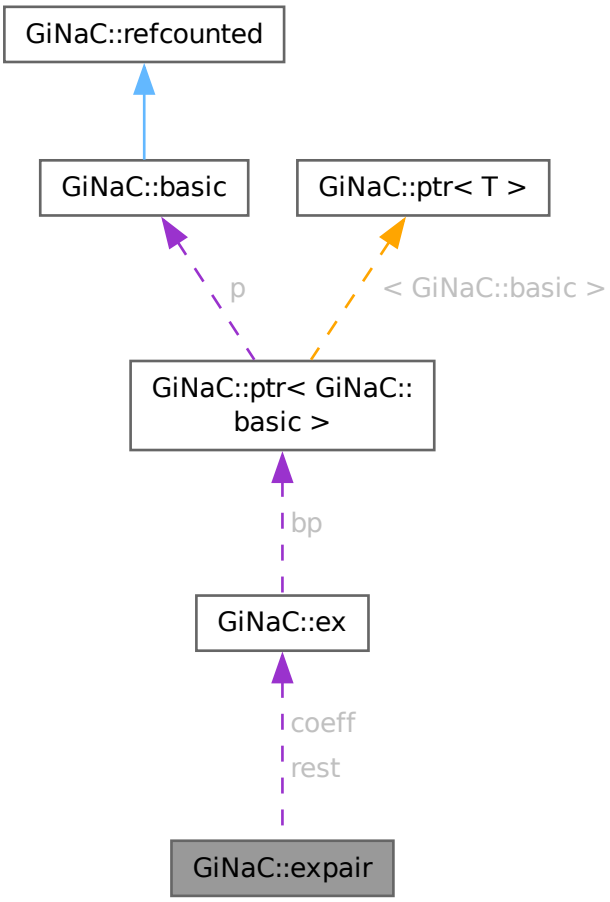
- ex.h

8.53 GiNaC::expair Class Reference

A pair of expressions.

```
#include <expair.h>
```

Collaboration diagram for GiNaC::expair:



Public Member Functions

- expair ()
- expair (const ex &r, const ex &c)

  Construct an expair from two ex.
- bool is_equal (const expair &other) const

  Member-wise check for canonical ordering equality.
- bool is_less (const expair &other) const

  Member-wise check for canonical ordering lessness.
- int compare (const expair &other) const

  Member-wise check for canonical ordering.
- void print (std::ostream &os) const
- bool is_canonical_numeric () const

  True if this is of the form (numeric,ex(1)).
- void swap (expair &other)

  Swap contents with other expair.
- const expair conjugate () const

Public Attributes

- ex rest

  first member of pair, an arbitrary expression
- ex coeff

  second member of pair, must be numeric

8.53.1 Detailed Description

A pair of expressions.

This is similar to STL's pair<>. It is slightly extended since we need to account for methods like .compare(). Also, since this is meant for use by class expairseq it must satisfy the invariance that the member coeff must be of type numeric.

8.53.2 Constructor & Destructor Documentation

8.53.2.1 expair() [1/2]

```
GiNaC::expair::expair ()  [inline]
```

References coeff, and rest.

Referenced by GiNaC::pseries::add_series(), GiNaC::atan_series(), GiNaC::atanh_series(), GiNaC::add::coeff(), GiNaC::add::combine_ex_with_coeff_to_pair(), GiNaC::expairseq::combine_ex_with_coeff_to_pair(), GiNaC::mul::combine_ex_with_
GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::expairseq::combine_pair_with_coeff_to_pair(), GiNaC::mul::combine_pair_w
compare(), conjugate(), GiNaC::expairseq::construct_from_2_expairseq(), GiNaC::expairseq::construct_from_expairseq_ex(), GiNaC::csgn_series(), GiNaC::add::derivative(), GiNaC::power::derivative(), GiNaC::pseries::derivative(), GiNaC::mul::do_print_latex(), GiNaC::EllipticE_series(), GiNaC::EllipticK_series(), GiNaC::eta_series(), GiNaC::mul::eval(), GiNaC::pseries::eval_integ(), GiNaC::mul::evalf(), GiNaC::pseries::evalf(), GiNaC::pseries::evalm(), GiNaC::mul::expand(), GiNaC::pseries::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), GiNaC::expairseq::expandchildren(), GiNaC::H_series(), GiNaC::add::imag_part(), GiNaC::pseries::imag_part(), is_equal(), is_less(), GiNaC::Li2_series(), GiNaC::Li_series(), GiNaC::log_series(), GiNaC::expairseq::make_flat(), GiNaC::pseries::mul_const(), GiNaC::pseries::mul_series(), GiNaC::pseries::normal(), GiNaC::Order_series(), GiNaC::pseries::power_const(), GiNaC::expairseq::read_archive(), GiNaC::pseries::read_archive(), GiNaC::add::real_part(), GiNaC::pseries::real_part(), GiNaC::S_series(), GiNaC::basic::series(), GiNaC::integral::series(), GiNaC::power::series(), GiNaC::pseries::series(), GiNaC::symbol::series(), GiNaC::add::smod(), GiNaC::add::split_ex_to_pair(), GiNaC::expairseq::split_ex_to_pair(), GiNaC::mul::split_ex_to_pair(), GiNaC::sqrfree_yun(), GiNaC::step_series(), GiNaC::pseries::subs(), swap(), GiNaC::expairseq::to_polynomial(), and GiNaC::expairseq::to_rational().

8.53.2.2 expair() [2/2]

```
GiNaC::expair::expair (
            const ex & r,
            const ex & c)  [inline]
```

Construct an expair from two ex.

References coeff, GINAC_ASSERT, GiNaC::is_exactly_a(), and rest.

8.53.3 Member Function Documentation

8.53.3.1 is_equal()

```
bool GiNaC::expair::is_equal (
            const expair & other) const  [inline]
```

Member-wise check for canonical ordering equality.

References coeff, expair(), GiNaC::ex::is_equal(), and rest.

Referenced by GiNaC::conjugateepvector(), GiNaC::expairseq::evalchildren(), GiNaC::mul::expair_needs_further_processing(), and GiNaC::expairseq::subschildren().

8.53.3.2 is_less()

```
bool GiNaC::expair::is_less (
            const expair & other) const  [inline]
```

Member-wise check for canonical ordering lessness.

References coeff, GiNaC::ex::compare(), expair(), and rest.

Referenced by GiNaC::expair_is_less::operator()().

8.53.3.3 compare()

```
int GiNaC::expair::compare (
            const expair & other) const  [inline]
```

Member-wise check for canonical ordering.

References coeff, GiNaC::ex::compare(), expair(), and rest.

8.53.3.4 print()

```
void GiNaC::expair::print (
            std::ostream & os) const
```

References coeff, GiNaC::print_tree::delta_indent, GiNaC::ex::print(), and rest.

8.53.3.5 is_canonical_numeric()

```
bool GiNaC::expair::is_canonical_numeric () const  [inline]
```

True if this is of the form (numeric,ex(1)).

References coeff, GINAC_ASSERT, GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), and rest.

8.53.3.6 swap()

```
void GiNaC::expair::swap (
            expair & other)  [inline]
```

Swap contents with other expair.

References coeff, expair(), rest, and GiNaC::ex::swap().

Referenced by GiNaC::mul::derivative(), GiNaC::expair_swap::operator()(), and GiNaC::swap().

8.53.3.7 conjugate()

```
const expair GiNaC::expair::conjugate () const
```

References GiNaC::are_ex_trivially_equal(), coeff, GiNaC::ex::conjugate(), expair(), and rest.

Referenced by GiNaC::conjugateepvector().

8.53.4 Member Data Documentation

8.53.4.1 rest

```
ex GiNaC::expair::rest
```

first member of pair, an arbitrary expression

Referenced by GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::expairseq::combine_pair_with_coeff_to_pair(), GiNaC::mul::combine_pair_with_coeff_to_pair(), compare(), conjugate(), GiNaC::expairseq::construct_from_2_ex(), GiNaC::expairseq::construct_from_expairseq_ex(), expair(), expair(), is_canonical_numeric(), is_equal(), is_less(), GiNaC::expair_rest_is_less::operator()(), print(), GiNaC::expairseq::printpair(), GiNaC::add::recombine_pair_to_ex(), GiNaC::expairseq::recombine_pair_to_ex(), GiNaC::mul::recombine_pair_to_ex(), and swap().

8.53.4.2 coeff

```
ex GiNaC::expair::coeff
```

second member of pair, must be numeric

Referenced by GiNaC::mul::can_make_flat(), GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::expairseq::combine_pair_with_
GiNaC::mul::combine_pair_with_coeff_to_pair(), compare(), conjugate(), GiNaC::expairseq::construct_from_2_ex(), GiNaC::expairseq::construct_from_expairseq_ex(), expair(), expair(), GiNaC::power::expand_mul(), is_canonical_numeric(), is_equal(), is_less(), print(), GiNaC::expairseq::printpair(), GiNaC::add::recombine_pair_to_ex(), GiNaC::expairseq::recombine_pair_t
GiNaC::mul::recombine_pair_to_ex(), and swap().

The documentation for this class was generated from the following files:

- expair.h
- expair.cpp

8.54 GiNaC::expair_is_less Struct Reference

Function object for insertion into third argument of STL's sort() etc.

`#include <expair.h>`

Public Member Functions

- bool operator() (const expair &lh, const expair &rh) const

8.54.1 Detailed Description

Function object for insertion into third argument of STL's sort() etc.

8.54.2 Member Function Documentation

8.54.2.1 operator()()

```
bool GiNaC::expair_is_less::operator() (
            const expair & lh,
            const expair & rh) const  [inline]
```

References GiNaC::expair::is_less().

The documentation for this struct was generated from the following file:

- expair.h

8.55 GiNaC::expair_rest_is_less Struct Reference

Function object not caring about the numerical coefficients for insertion into third argument of STL's sort().

`#include <expair.h>`

Public Member Functions

- bool operator() (const expair &lh, const expair &rh) const

8.55.1 Detailed Description

Function object not caring about the numerical coefficients for insertion into third argument of STL's sort().

Note that this does not define a strict weak ordering since for any symbol x we have neither $3*x<2*x$ or $2*x<3*x$. Handle with care!

8.55.2 Member Function Documentation

8.55.2.1 operator()()

```
bool GiNaC::expair_rest_is_less::operator() (
            const expair & lh,
            const expair & rh) const  [inline]
```

References GiNaC::ex::compare(), and GiNaC::expair::rest.

The documentation for this struct was generated from the following file:

- expair.h

8.56 GiNaC::expair_swap Struct Reference

```
#include <expair.h>
```

Public Member Functions

- void operator() (expair &lh, expair &rh) const

8.56.1 Member Function Documentation

8.56.1.1 operator()()

```
void GiNaC::expair_swap::operator() (
            expair & lh,
            expair & rh) const  [inline]
```

References GiNaC::expair::swap().

The documentation for this struct was generated from the following file:

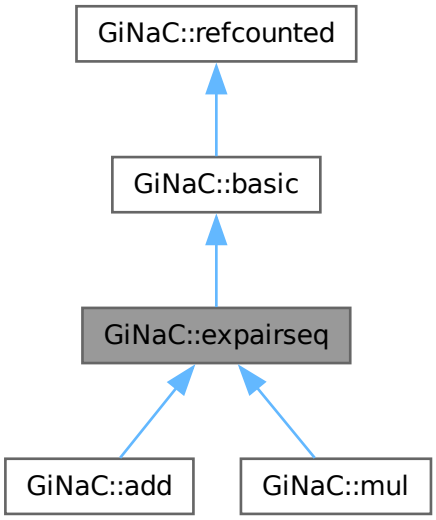
- expair.h

8.57 GiNaC::expairseq Class Reference

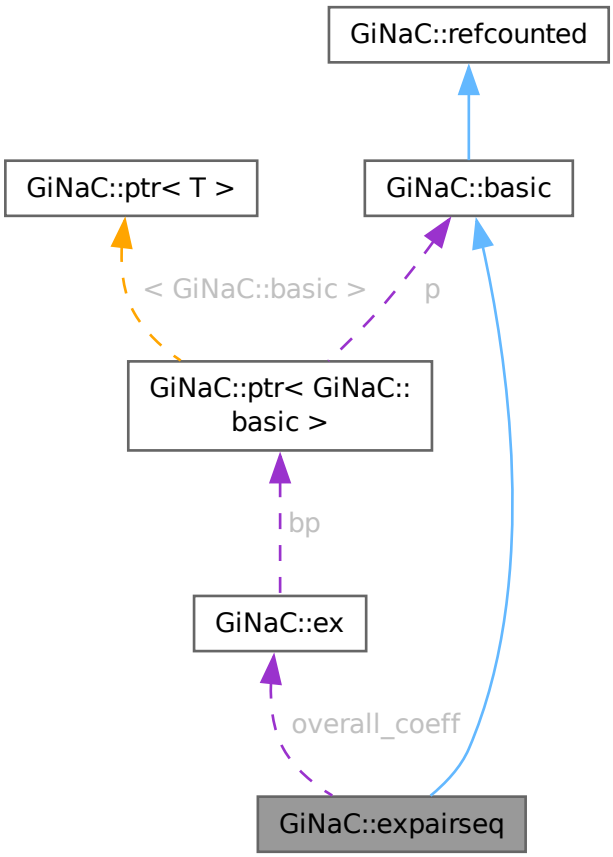
A sequence of class expair.

`#include <expairseq.h>`

Inheritance diagram for GiNaC::expairseq:



Collaboration diagram for GiNaC::expairseq:



Public Member Functions

- expairseq (const ex &lh, const ex &rh)
- expairseq (const exvector &v)
- expairseq (const epvector &v, const ex &oc, bool do_index_renaming=false)
- expairseq (epvector &&vp, const ex &oc, bool do_index_renaming=false)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex map (map_function &f) const override

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- ex eval () const override

  Perform coefficient-wise automatic term rewriting rules in this class.

- ex to_rational (exmap &repl) const override

  Implementation of ex::to_rational() for expairseqs.
- ex to_polynomial (exmap &repl) const override

  Implementation of ex::to_polynomial() for expairseqs.
- bool match (const ex &pattern, exmap &repl_lst) const override

  Check whether the expression matches a given pattern.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex conjugate () const override
- void archive (archive_node &n) const override

  Save (serialize) the object into archive node.
- void read_archive (const archive_node &n, lst &syms) override

  Load (deserialize) the object from an archive node.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.

- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual return_type_t return_type_tinfo () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- unsigned return_type () const override
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- virtual ex thisexpairseq (const epvector &v, const ex &oc, bool do_index_renaming=false) const

  Create an object of this type.
- virtual ex thisexpairseq (epvector &&vp, const ex &oc, bool do_index_renaming=false) const
- virtual void printseq (const print_context &c, char delim, unsigned this_precedence, unsigned upper_precedence) const
- virtual void printpair (const print_context &c, const expair &p, unsigned upper_precedence) const
- virtual expair split_ex_to_pair (const ex &e) const

  Form an expair from an ex, using the corresponding semantics.
- virtual expair combine_ex_with_coeff_to_pair (const ex &e, const ex &c) const
- virtual expair combine_pair_with_coeff_to_pair (const expair &p, const ex &c) const
- virtual ex recombine_pair_to_ex (const expair &p) const

  Form an ex out of an expair, using the corresponding semantics.
- virtual bool expair_needs_further_processing (epp it)
- virtual ex default_overall_coeff () const
- virtual void combine_overall_coeff (const ex &c)
- virtual void combine_overall_coeff (const ex &c1, const ex &c2)
- virtual bool can_make_flat (const expair &p) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void construct_from_2_ex (const ex &lh, const ex &rh)
- void construct_from_2_expairseq (const expairseq &s1, const expairseq &s2)
- void construct_from_expairseq_ex (const expairseq &s, const ex &e)
- void construct_from_exvector (const exvector &v)
- void construct_from_epvector (const epvector &v, bool do_index_renaming=false)
- void construct_from_epvector (epvector &&v, bool do_index_renaming=false)
- void make_flat (const exvector &v)

  Combine this expairseq with argument exvector.
- void make_flat (const epvector &v, bool do_index_renaming=false)

  Combine this expairseq with argument epvector.
- void canonicalize ()

  Brings this expairseq into a sorted (canonical) form.
- void combine_same_terms_sorted_seq ()

  Compact a presorted expairseq by combining all matching expairs to one each.
- bool is_canonical () const

  Check if this expairseq is in sorted (canonical) form.
- epvector expandchildren (unsigned options) const

  Member-wise expand the expairs in this sequence.
- epvector evalchildren () const

  Member-wise evaluate the expairs in this sequence.
- epvector subschildren (const exmap &m, unsigned options=0) const

  Member-wise substitute in this sequence.

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- epvector seq
- ex overall_coeff

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.57.1 Detailed Description

A sequence of class expair.

This is used for time-critical classes like sums and products of terms since handling a list of coeff and rest is much faster than handling a list of products or powers, respectively. (Not incidentally, Maple does it the same way, maybe others too.) The semantics is (at least) twofold: one for addition and one for multiplication and several methods have to be overridden by derived classes to reflect the change in semantics. However, most functionality turns out to be shared between addition and multiplication, which is the reason why there is this base class.

8.57.2 Constructor & Destructor Documentation

8.57.2.1 expairseq() [1/4]

```
GiNaC::expairseq::expairseq (
            const ex & lh,
            const ex & rh)
```

References construct_from_2_ex(), GiNaC::basic::ex, GINAC_ASSERT, and is_canonical().

Referenced by construct_from_2_expairseq(), construct_from_expairseq_ex(), is_equal_same_type(), make_flat(), make_flat(), thisexpairseq(), and thisexpairseq().

8.57.2.2 expairseq() [2/4]

```
GiNaC::expairseq::expairseq (
            const exvector & v)
```

References construct_from_exvector(), GINAC_ASSERT, and is_canonical().

8.57.2.3 expairseq() [3/4]

```
GiNaC::expairseq::expairseq (
            const epvector & v,
            const ex & oc,
            bool do_index_renaming = false)
```

References construct_from_epvector(), GiNaC::basic::ex, GINAC_ASSERT, GiNaC::is_a(), is_canonical(), and overall_coeff.

8.57.2.4 expairseq() [4/4]

```
GiNaC::expairseq::expairseq (
            epvector && vp,
            const ex & oc,
            bool do_index_renaming = false)
```

References construct_from_epvector(), GiNaC::basic::ex, GINAC_ASSERT, GiNaC::is_a(), is_canonical(), and overall_coeff.

8.57.3 Member Function Documentation

8.57.3.1 precedence()

```
unsigned GiNaC::expairseq::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::mul.

Referenced by do_print(), and printpair().

8.57.3.2 info()

```
bool GiNaC::expairseq::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::mul.

References GiNaC::basic::clearflag(), GiNaC::info_flags::expanded, GiNaC::status_flags::expanded, GiNaC::basic::flags, GiNaC::info_flags::has_indices, GiNaC::status_flags::has_indices, GiNaC::status_flags::has_no_indices, seq, and GiNaC::basic::setflag().

8.57.3.3 nops()

```
size_t GiNaC::expairseq::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

References default_overall_coeff(), GiNaC::ex::is_equal(), overall_coeff, and seq.

Referenced by GiNaC::algebraic_match_mul_with_mul(), GiNaC::mul::algebraic_subs_mul(), GiNaC::add::conjugate(), GiNaC::add::do_print_python_repr(), GiNaC::mul::do_print_python_repr(), do_print_tree(), GiNaC::mul::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), GiNaC::add::get_free_indices(), GiNaC::mul::get_free_indices(), GiNaC::mul::has(), match(), and GiNaC::Order_eval().

8.57.3.4 op()

```
ex GiNaC::expairseq::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References default_overall_coeff(), GiNaC::basic::ex, GINAC_ASSERT, GiNaC::ex::is_equal(), overall_coeff, recombine_pair_to_ex(), and seq.

Referenced by GiNaC::algebraic_match_mul_with_mul(), GiNaC::mul::algebraic_subs_mul(), GiNaC::add::conjugate(), GiNaC::add::do_print_python_repr(), GiNaC::mul::do_print_python_repr(), GiNaC::add::get_free_indices(), GiNaC::mul::get_free_indices(), match(), GiNaC::Order_eval(), GiNaC::add::series(), and GiNaC::mul::series().

8.57.3.5 map()

```
ex GiNaC::expairseq::map (
            map_function & f) const  [override], [virtual]
```

Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

Reimplemented from GiNaC::basic.

References default_overall_coeff(), GiNaC::basic::ex, GiNaC::is_a(), GiNaC::ex::is_equal(), overall_coeff, recombine_pair_to_ex(), seq, split_ex_to_pair(), and thisexpairseq().

8.57.3.6 eval()

```
ex GiNaC::expairseq::eval () const  [override], [virtual]
```

Perform coefficient-wise automatic term rewriting rules in this class.

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::mul.

References GiNaC::dynallocate(), evalchildren(), GiNaC::status_flags::evaluated, GiNaC::basic::ex, GiNaC::basic::flags, GiNaC::basic::hold(), and overall_coeff.

8.57.3.7 to_rational()

```
ex GiNaC::expairseq::to_rational (
            exmap & repl) const  [override], [virtual]
```

Implementation of ex::to_rational() for expairseqs.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, default_overall_coeff(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::ex::info(), GiNaC::info_flags::numeric, overall_coeff, recombine_pair_to_ex(), seq, split_ex_to_pair(), thisexpairseq(), GiNaC::ex::to_rational(), and to_rational().

Referenced by to_rational().

8.57.3.8 to_polynomial()

```
ex GiNaC::expairseq::to_polynomial (
            exmap & repl) const  [override], [virtual]
```

Implementation of ex::to_polynomial() for expairseqs.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, default_overall_coeff(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::ex::info(), GiNaC::info_flags::numeric, overall_coeff, recombine_pair_to_ex(), seq, split_ex_to_pair(), thisexpairseq(), GiNaC::ex::to_polynomial(), and to_polynomial().

Referenced by to_polynomial().

8.57.3.9 match()

```
bool GiNaC::expairseq::match (
            const ex & pattern,
            exmap & repl_lst) const  [override], [virtual]
```

Check whether the expression matches a given pattern.

For every wildcard object in the pattern, a pair with the wildcard as a key and matching expression as a value is added to repl_lst.

Reimplemented from GiNaC::basic.

References default_overall_coeff(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::ex::nops(), nops(), GiNaC::ex::op(), op(), split_ex_to_pair(), and thisexpairseq().

8.57.3.10 subs()

```
ex GiNaC::expairseq::subs (
            const exmap & m,
            unsigned options = 0) const  [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References GiNaC::subs_options::algebraic, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::is_exactly_a(), GiNaC::subs_options::no_inde overall_coeff, GiNaC::basic::subs_one_level(), subschildren(), and thisexpairseq().

8.57.3.11 conjugate()

```
ex GiNaC::expairseq::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::mul.

References GiNaC::are_ex_trivially_equal(), GiNaC::ex::conjugate(), GiNaC::conjugateepvector(), GiNaC::basic::ex, overall_coeff, seq, and thisexpairseq().

8.57.3.12 archive()

```
void GiNaC::expairseq::archive (
            archive_node & n) const  [override], [virtual]
```

Save (serialize) the object into archive node.

Archive the object.

Losely speaking, this method turns an expression into a byte stream (which can be saved and restored later on, or sent via network, etc.)

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_ex(), overall_coeff, and seq.

8.57.3.13 read_archive()

```
void GiNaC::expairseq::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented from GiNaC::basic.

References canonicalize(), GiNaC::basic::coeff(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::archive_node::find_ex(), GiNaC::archive_node::find_ex_by_loc(), GiNaC::archive_node::find_property_range(), GINAC_ASSERT, is_canonical(), overall_coeff, and seq.

8.57.3.14 is_equal_same_type()

```
bool GiNaC::expairseq::is_equal_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if two objects of same type are equal.

Normally needs not be reimplemented as long as it wasn't overwritten by some parent class, since it just calls compare_same_type(). The reason why this function exists is that sometimes it is easier to determine equality than an order relation and then it can be overridden.

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic(), expairseq(), GiNaC::ex::is_equal(), overall_coeff, and seq.

8.57.3.15 return_type()

```
unsigned GiNaC::expairseq::return_type () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::mul.

References GiNaC::return_types::noncommutative_composite.

8.57.3.16 calchash()

```
unsigned GiNaC::expairseq::calchash () const  [override], [protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References GiNaC::status_flags::evaluated, GiNaC::basic::flags, GiNaC::ex::gethash(), GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, overall_coeff, GiNaC::rotate_left(), seq, and GiNaC::basic::setflag().

8.57.3.17 expand()

```
ex GiNaC::expairseq::expand (
            unsigned options = 0) const  [override], [protected], [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::mul.

References GiNaC::basic::ex, expandchildren(), GiNaC::status_flags::expanded, overall_coeff, GiNaC::basic::setflag(), and thisexpairseq().

8.57.3.18 thisexpairseq() [1/2]

```
ex GiNaC::expairseq::thisexpairseq (
             const epvector & v,
             const ex & oc,
             bool do_index_renaming = false) const  [protected], [virtual]
```

Create an object of this type.

This method works similar to a constructor. It is useful because expairseq has (at least) two possible different semantics but we want to inherit methods thus avoiding code duplication. Sometimes a method in expairseq has to create a new one of the same semantics, which cannot be done by a ctor because the name (add, mul,...) is unknown on the expairseq level. In order for this trick to work a derived class must of course override this definition.

Reimplemented in GiNaC::add, and GiNaC::mul.

References GiNaC::basic::ex, and expairseq().

Referenced by conjugate(), expand(), map(), match(), subs(), to_polynomial(), and to_rational().

8.57.3.19 thisexpairseq() [2/2]

```
ex GiNaC::expairseq::thisexpairseq (
             epvector && vp,
             const ex & oc,
             bool do_index_renaming = false) const  [protected], [virtual]
```

Reimplemented in GiNaC::add, and GiNaC::mul.

References GiNaC::basic::ex, and expairseq().

8.57.3.20 printseq()

```
void GiNaC::expairseq::printseq (
             const print_context & c,
             char delim,
             unsigned this_precedence,
             unsigned upper_precedence) const  [protected], [virtual]
```

References default_overall_coeff(), GiNaC::ex::is_equal(), overall_coeff, GiNaC::ex::print(), printpair(), GiNaC::print_context::s, and seq.

Referenced by do_print().

8.57.3.21 printpair()

```
void GiNaC::expairseq::printpair (
             const print_context & c,
             const expair & p,
             unsigned upper_precedence) const  [protected], [virtual]
```

References GiNaC::expair::coeff, precedence(), GiNaC::ex::print(), GiNaC::expair::rest, and GiNaC::print_context::s.

Referenced by is_canonical(), and printseq().

8.57.3.22 split_ex_to_pair()

```
expair GiNaC::expairseq::split_ex_to_pair (
            const ex & e) const  [protected], [virtual]
```

Form an expair from an ex, using the corresponding semantics.

See also

expairseq::recombine_pair_to_ex()

Reimplemented in GiNaC::add, and GiNaC::mul.

References GiNaC::_ex1, GiNaC::basic::ex, and GiNaC::expair::expair().

Referenced by construct_from_2_ex(), construct_from_expairseq_ex(), make_flat(), map(), match(), subschildren(), to_polynomial(), and to_rational().

8.57.3.23 combine_ex_with_coeff_to_pair()

```
expair GiNaC::expairseq::combine_ex_with_coeff_to_pair (
            const ex & e,
            const ex & c) const  [protected], [virtual]
```

Reimplemented in GiNaC::add, and GiNaC::mul.

References GiNaC::basic::ex, GiNaC::expair::expair(), GINAC_ASSERT, and GiNaC::is_exactly_a().

Referenced by evalchildren(), and subschildren().

8.57.3.24 combine_pair_with_coeff_to_pair()

```
expair GiNaC::expairseq::combine_pair_with_coeff_to_pair (
            const expair & p,
            const ex & c) const  [protected], [virtual]
```

Reimplemented in GiNaC::add, and GiNaC::mul.

References GiNaC::expair::coeff, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, GiNaC::is_exactly_a(), and GiNaC::expair::rest.

8.57.3.25 recombine_pair_to_ex()

```
ex GiNaC::expairseq::recombine_pair_to_ex (
            const expair & p) const  [protected], [virtual]
```

Form an ex out of an expair, using the corresponding semantics.

See also

expairseq::split_ex_to_pair()

Reimplemented in GiNaC::add, and GiNaC::mul.

References GiNaC::expair::coeff, GiNaC::basic::ex, and GiNaC::expair::rest.

Referenced by map(), op(), subschildren(), to_polynomial(), and to_rational().

8.57.3.26 expair_needs_further_processing()

```
bool GiNaC::expairseq::expair_needs_further_processing (
            epp it)  [protected], [virtual]
```

Reimplemented in GiNaC::mul.

References GiNaC::_ex1, and GiNaC::is_exactly_a().

Referenced by combine_same_terms_sorted_seq(), construct_from_2_expairseq(), and construct_from_expairseq_ex().

8.57.3.27 default_overall_coeff()

```
ex GiNaC::expairseq::default_overall_coeff () const  [protected], [virtual]
```

Reimplemented in GiNaC::mul.

References GiNaC::_ex0, and GiNaC::basic::ex.

Referenced by do_print_tree(), map(), match(), nops(), op(), printseq(), to_polynomial(), and to_rational().

8.57.3.28 combine_overall_coeff() [1/2]

```
void GiNaC::expairseq::combine_overall_coeff (
            const ex & c)  [protected], [virtual]
```

Reimplemented in GiNaC::mul.

References GiNaC::basic::ex, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), and overall_coeff.

Referenced by construct_from_2_ex(), construct_from_2_expairseq(), construct_from_expairseq_ex(), make_flat(), and make_flat().

8.57.3.29 combine_overall_coeff() [2/2]

```
void GiNaC::expairseq::combine_overall_coeff (
            const ex & c1,
            const ex & c2)  [protected], [virtual]
```

Reimplemented in GiNaC::mul.

References GiNaC::basic::ex, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::mul::mul(), and overall_coeff.

8.57.3.30 can_make_flat()

```
bool GiNaC::expairseq::can_make_flat (
            const expair & p) const  [protected], [virtual]
```

Reimplemented in GiNaC::mul.

8.57.3.31 do_print()

```
void GiNaC::expairseq::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References precedence(), printseq(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.57.3.32 do_print_tree()

```
void GiNaC::expairseq::do_print_tree (
            const print_tree & c,
            unsigned level) const  [protected]
```

References default_overall_coeff(), GiNaC::print_tree::delta_indent, GiNaC::basic::flags, GiNaC::basic::hashvalue, GiNaC::ex::is_equal(), nops(), overall_coeff, GiNaC::ex::print(), GiNaC::print_context::s, and seq.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.57.3.33 construct_from_2_ex()

```
void GiNaC::expairseq::construct_from_2_ex (
            const ex & lh,
            const ex & rh)  [protected]
```

References GiNaC::expair::coeff, combine_overall_coeff(), GiNaC::ex::compare(), construct_from_2_expairseq(), construct_from_expairseq_ex(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::info_flags::has_indices, GiNaC::ex::info(), GiNaC::is_a(), GiNaC::is_exactly_a(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::expair::rest, seq, and split_ex_to_pair().

Referenced by GiNaC::add::add(), expairseq(), and GiNaC::mul::mul().

8.57.3.34 construct_from_2_expairseq()

```
void GiNaC::expairseq::construct_from_2_expairseq (
            const expairseq & s1,
            const expairseq & s2)  [protected]
```

References GiNaC::add::add(), combine_overall_coeff(), construct_from_epvector(), GiNaC::ex_to(), GiNaC::expair::expair(), expair_needs_further_processing(), expairseq(), GiNaC::numeric::is_zero(), overall_coeff, and seq.

Referenced by construct_from_2_ex().

8.57.3.35 construct_from_expairseq_ex()

```
void GiNaC::expairseq::construct_from_expairseq_ex (
            const expairseq & s,
            const ex & e)  [protected]
```

References GiNaC::add::add(), GiNaC::class_info< OPT >::first, GiNaC::expair::coeff, combine_overall_coeff(), construct_from_epvector(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), expair_needs_further_processing(), expairseq(), GiNaC::is_exactly_a(), GiNaC::numeric::is_zero(), overall_coeff, GiNaC::expair::rest, seq, and split_ex_to_pair().

Referenced by construct_from_2_ex().

8.57.3.36 construct_from_exvector()

```
void GiNaC::expairseq::construct_from_exvector (
            const exvector & v)  [protected]
```

References canonicalize(), combine_same_terms_sorted_seq(), and make_flat().

Referenced by GiNaC::add::add(), expairseq(), GiNaC::mul::mul(), and GiNaC::mul::mul().

8.57.3.37 construct_from_epvector() [1/2]

```
void GiNaC::expairseq::construct_from_epvector (
            const epvector & v,
            bool do_index_renaming = false)  [protected]
```

References canonicalize(), combine_same_terms_sorted_seq(), and make_flat().

Referenced by GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), combine_same_terms_sorted_seq(), construct_from_2_expairseq(), construct_from_expairseq_ex(), expairseq(), expairseq(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), and GiNaC::mul::mul().

8.57.3.38 construct_from_epvector() [2/2]

```
void GiNaC::expairseq::construct_from_epvector (
            epvector && v,
            bool do_index_renaming = false)  [protected]
```

References canonicalize(), combine_same_terms_sorted_seq(), and make_flat().

8.57.3.39 make_flat() [1/2]

```
void GiNaC::expairseq::make_flat (
            const exvector & v)  [protected]
```

Combine this expairseq with argument exvector.

It cares for associativity as well as for special handling of numerics.

References GiNaC::_ex1, combine_overall_coeff(), GiNaC::basic::ex, GiNaC::ex_to(), expairseq(), GiNaC::make_flat_inserter::handle
GiNaC::info_flags::has_indices, GiNaC::is_a(), GiNaC::is_exactly_a(), overall_coeff, seq, and split_ex_to_pair().

Referenced by construct_from_epvector(), construct_from_epvector(), and construct_from_exvector().

8.57.3.40 make_flat() [2/2]

```
void GiNaC::expairseq::make_flat (
            const epvector & v,
            bool do_index_renaming = false)  [protected]
```

Combine this expairseq with argument epvector.

It cares for associativity as well as for special handling of numerics.

References GiNaC::_ex1, GiNaC::are_ex_trivially_equal(), GiNaC::ex::coeff(), combine_overall_coeff(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), expairseq(), GiNaC::make_flat_inserter::handle_factor(), GiNaC::info_flags::has_indices, GiNaC::is_a(), overall_coeff, and seq.

8.57.3.41 canonicalize()

```
void GiNaC::expairseq::canonicalize ()  [protected]
```

Brings this expairseq into a sorted (canonical) form.

References seq.

Referenced by construct_from_epvector(), construct_from_epvector(), construct_from_exvector(), and read_archive().

8.57.3.42 combine_same_terms_sorted_seq()

```
void GiNaC::expairseq::combine_same_terms_sorted_seq ()  [protected]
```

Compact a presorted expairseq by combining all matching expairs to one each.

On an add object, this is responsible for 2∗x+3∗x+y -> 5∗x+y, for instance.

References construct_from_epvector(), GiNaC::ex_to(), expair_needs_further_processing(), and seq.

Referenced by construct_from_epvector(), construct_from_epvector(), and construct_from_exvector().

8.57.3.43 is_canonical()

```
bool GiNaC::expairseq::is_canonical () const  [protected]
```

Check if this expairseq is in sorted (canonical) form.

Useful mainly for debugging or in assertions since being sorted is an invariance.

References GiNaC::is_exactly_a(), GiNaC::print_tree::print_tree(), printpair(), and seq.

Referenced by GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), expairseq(), expairseq(), expairseq(), expairseq(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), and read_archive().

8.57.3.44 expandchildren()

```
epvector GiNaC::expairseq::expandchildren (
            unsigned options) const  [protected]
```

Member-wise expand the expairs in this sequence.

See also

expairseq::expand()

Returns

epvector containing expanded pairs, empty if no members had to be changed.

References GiNaC::are_ex_trivially_equal(), GiNaC::ex::coeff(), GiNaC::basic::ex, GiNaC::expair::expair(), and seq.

Referenced by GiNaC::add::expand(), and expand().

8.57.3.45 evalchildren()

```
epvector GiNaC::expairseq::evalchildren () const  [protected]
```

Member-wise evaluate the expairs in this sequence.

See also

expairseq::eval()

Returns

epvector containing evaluated pairs, empty if no members had to be changed.

References combine_ex_with_coeff_to_pair(), GiNaC::expair::is_equal(), seq, and unlikely.

Referenced by GiNaC::add::eval(), eval(), and GiNaC::mul::eval().

8.57.3.46 subschildren()

```
epvector GiNaC::expairseq::subschildren (
            const exmap & m,
            unsigned options = 0) const  [protected]
```

Member-wise substitute in this sequence.

See also

expairseq::subs()

Returns

epvector containing expanded pairs, empty if no members had to be changed.

References GiNaC::are_ex_trivially_equal(), GiNaC::ex::coeff(), combine_ex_with_coeff_to_pair(), GiNaC::basic::ex, GiNaC::expair::is_equal(), GiNaC::is_exactly_a(), GiNaC::subs_options::pattern_is_not_product, GiNaC::subs_options::pattern_is_pr recombine_pair_to_ex(), seq, split_ex_to_pair(), and GiNaC::ex::subs().

Referenced by subs().

8.57.4 Member Data Documentation

8.57.4.1 seq

```
epvector GiNaC::expairseq::seq  [protected]
```

Referenced by archive(), calchash(), GiNaC::mul::can_be_further_expanded(), canonicalize(), GiNaC::add::coeff(), GiNaC::mul::coeff(), combine_same_terms_sorted_seq(), conjugate(), GiNaC::mul::conjugate(), construct_from_2_ex(), construct_from_2_expairseq(), construct_from_expairseq_ex(), GiNaC::add::degree(), GiNaC::mul::degree(), GiNaC::add::derivative(), GiNaC::mul::derivative(), GiNaC::mul::do_print(), GiNaC::add::do_print_csrc(), GiNaC::mul::do_print_csrc(), GiNaC::mul::do_print_latex(), do_print_tree(), GiNaC::add::eval(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::add::eval_ncmul(), GiNaC::mul::eval_ncmul(), evalchildren(), GiNaC::mul::evalf(), GiNaC::add::evalm(), GiNaC::mul::evalm(), GiNaC::mul::expand(), GiNaC::power::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2() GiNaC::power::expand_mul(), expandchildren(), GiNaC::mul::expandchildren(), GiNaC::mul::find_real_imag(), GiNaC::add::imag_part(), GiNaC::add::info(), info(), GiNaC::mul::info(), GiNaC::add::integer_content(), GiNaC::mul::integer_content(), is_canonical(), is_equal_same_type(), GiNaC::add::is_polynomial(), GiNaC::mul::is_polynomial(), GiNaC::add::ldegree(), GiNaC::mul::ldegree(), make_flat(), make_flat(), map(), GiNaC::add::max_coefficient(), GiNaC::mul::max_coefficient(), nops(), GiNaC::add::normal(), GiNaC::mul::normal(), op(), GiNaC::add::print_add(), printseq(), read_archive(), GiNaC::add::real_part(), GiNaC::add::return_type(), GiNaC::mul::return_type(), GiNaC::add::return_type_tinfo(), GiNaC::mul::return_type_tinfo(), GiNaC::add::series(), GiNaC::mul::series(), GiNaC::add::smod(), GiNaC::mul::smod(), subschildren(), to_polynomial(), and to_rational().

8.57.4.2 overall_coeff

```
ex GiNaC::expairseq::overall_coeff  [protected]
```

Referenced by GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), archive(), calchash(), GiNaC::add::coeff(), GiNaC::mul::coeff(), GiNaC::add::combine_ex_with_coeff_to_pair(), combine_overall_coeff(), combine_overall_coeff(), GiNaC::mul::combine_overall_coeff(), GiNaC::mul::combine_overall_coeff(), conjugate(), GiNaC::mul::conjugate(), construct_from_2_expairseq(), construct_from_expairseq_ex(), GiNaC::add::degree(), GiNaC::mul::derivative(), GiNaC::add::do_print_csrc(), GiNaC::mul::do_print_csrc(), do_print_tree(), GiNaC::add::eval(), eval(), GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::mul::evalf(), GiNaC::add::evalm(), GiNaC::mul::evalm(), expairseq(), expairseq(), GiNaC::add::expand(), expand(), GiNaC::mul::expand(), GiNaC::power::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), GiNaC::power::expand_mul(), GiNaC::mul::find_real_imag(), GiNaC::add::imag_part(), GiNaC::add::info(), GiNaC::mul::info(), GiNaC::add::integer_content(), GiNaC::mul::integer_content(), is_equal_same_type(), GiNaC::add::ldegree(), make_flat(), make_flat(), map(), GiNaC::add::max_coefficient(), GiNaC::mul::max_coefficient(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), nops(), GiNaC::add::normal(), GiNaC::mul::normal(), op(), GiNaC::add::print_add(), GiNaC::mul::print_overall_coeff(), printseq(), read_archive(), GiNaC::add::real_part(), GiNaC::add::series(), GiNaC::mul::series(), GiNaC::add::smod(), GiNaC::mul::smod(), GiNaC::add::split_ex_to_pair(), subs(), to_polynomial(), and to_rational().

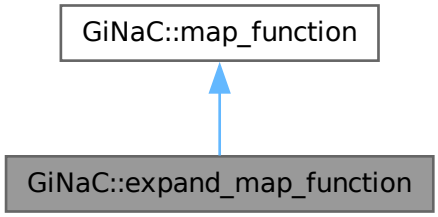
The documentation for this class was generated from the following files:

- expairseq.h
- expairseq.cpp
- normal.cpp

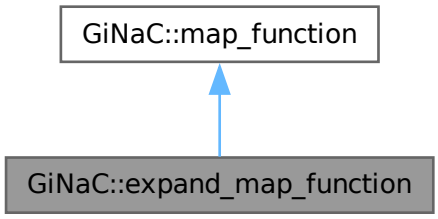
8.58 GiNaC::expand_map_function Struct Reference

Function object to be applied by basic::expand().

Inheritance diagram for GiNaC::expand_map_function:



Collaboration diagram for GiNaC::expand_map_function:



Public Member Functions

- expand_map_function (unsigned o)
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Public Attributes

- unsigned options

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.58.1 Detailed Description

Function object to be applied by basic::expand().

8.58.2 Constructor & Destructor Documentation

8.58.2.1 expand_map_function()

```
GiNaC::expand_map_function::expand_map_function (
            unsigned o)  [inline]
```

References options.

8.58.3 Member Function Documentation

8.58.3.1 operator()()

```
ex GiNaC::expand_map_function::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References GiNaC::ex::expand(), and options.

8.58.4 Member Data Documentation

8.58.4.1 options

```
unsigned GiNaC::expand_map_function::options
```

Referenced by expand_map_function(), and operator()().

The documentation for this struct was generated from the following file:

- basic.cpp

8.59 GiNaC::expand_options Class Reference

Flags to control the behavior of expand().

```
#include <flags.h>
```

Public Types

- enum { expand_indexed = 0x0001 , expand_function_args = 0x0002 , expand_rename_idx = 0x0004 , expand_transcendental = 0x0008 }

8.59.1 Detailed Description

Flags to control the behavior of expand().

8.59.2 Member Enumeration Documentation

8.59.2.1 anonymous enum

anonymous enum

Enumerator

expand_indexed	expands (a+b).i to a.i+b.i
expand_function_args	expands the arguments of functions
expand_rename_idx	used internally by mul::expand()
expand_transcendental	expands transcendental functions like log and exp

The documentation for this class was generated from the following file:

- flags.h

8.60 GiNaC::factor_options Class Reference

Flags to control the polynomial factorization.

#include <flags.h>

Public Types

- enum { polynomial = 0x0000 , all = 0x0001 }

8.60.1 Detailed Description

Flags to control the polynomial factorization.

8.60.2 Member Enumeration Documentation

8.60.2.1 anonymous enum

anonymous enum

Enumerator

polynomial	factor only expressions that are polynomials
all	factor all polynomial subexpressions

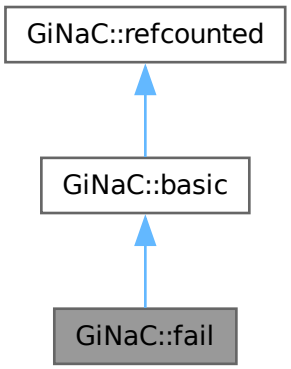
The documentation for this class was generated from the following file:

- flags.h

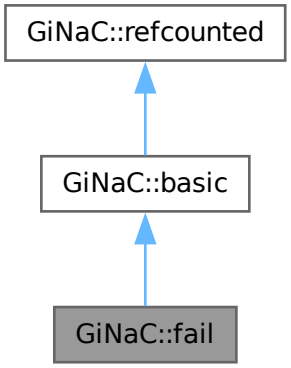
8.61 GiNaC::fail Class Reference

`#include <fail.h>`

Inheritance diagram for GiNaC::fail:



Collaboration diagram for GiNaC::fail:



Protected Member Functions

- unsigned return_type () const override
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.

- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.

- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.61.1 Member Function Documentation

8.61.1.1 return_type()

```
unsigned GiNaC::fail::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::noncommutative_composite.

8.61.1.2 do_print()

```
void GiNaC::fail::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

The documentation for this class was generated from the following file:

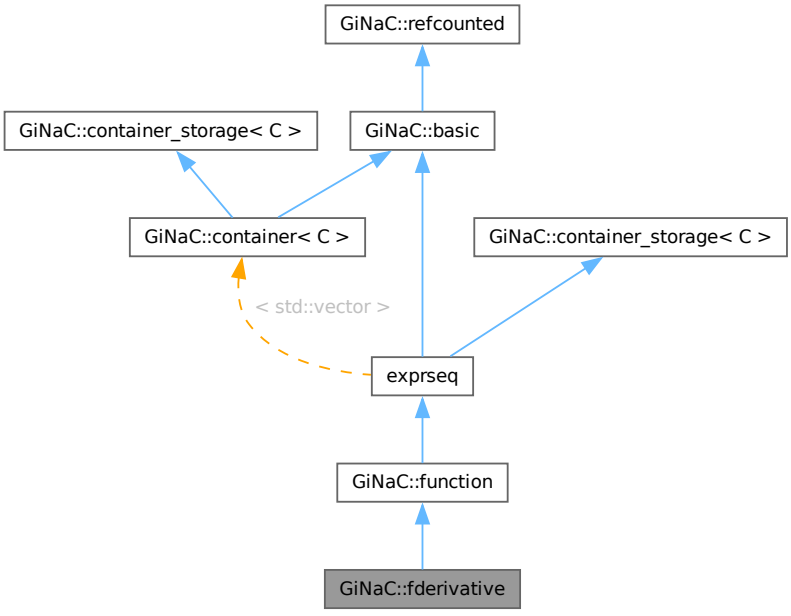
- fail.h

8.62 GiNaC::fderivative Class Reference

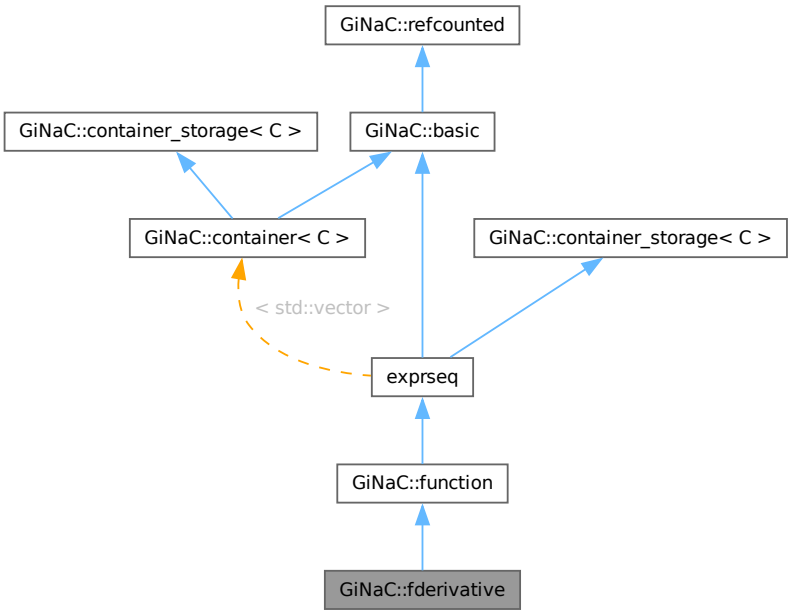
This class represents the (abstract) derivative of a symbolic function.

```
#include <fderivative.h>
```

Inheritance diagram for GiNaC::fderivative:



Collaboration diagram for GiNaC::fderivative:



Public Member Functions

- fderivative (unsigned ser, unsigned param, const exvector &args)

  Construct derivative with respect to one parameter.
- fderivative (unsigned ser, const paramset &params, const exvector &args)

  Construct derivative with respect to multiple parameters.
- fderivative (unsigned ser, const paramset &params, exvector &&v)
- void print (const print_context &c, unsigned level=0) const override

  Output to stream.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex series (const relational &r, int order, unsigned options=0) const override

  The series expansion of derivatives falls back to Taylor expansion.
- ex thiscontainer (const exvector &v) const override

  Similar to duplicate(), but with a preset sequence.
- ex thiscontainer (exvector &&v) const override

  Similar to duplicate(), but with a preset sequence (which gets pilfered).
- void archive (archive_node &n) const override

  Archive the object.
- void read_archive (const archive_node &n, lst &syms) override

  Load (deserialize) the object from an archive node.
- const paramset & derivatives () const

  Expose this object's derivative structure.

Public Member Functions inherited from GiNaC::function

- function (unsigned ser)
- function (unsigned ser, const ex &param1)
- function (unsigned ser, const ex &param1, const ex &param2)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10, const ex &param11)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10, const ex &param11, const ex &param12)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10, const ex &param11, const ex &param12, const ex &param13)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10, const ex &param11, const ex &param12, const ex &param13, const ex &param14)
- function (unsigned ser, const exprseq &es)
- function (unsigned ser, const exvector &v)
- function (unsigned ser, exvector &&v)
- void print (const print_context &c, unsigned level=0) const override

  Output to stream.
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex evalf () const override

  Evaluate object numerically.
- ex eval_ncmul (const exvector &v) const override

  This method is defined to be in line with behavior of function::return_type().
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- ex series (const relational &r, int order, unsigned options=0) const override

  Implementation of ex::series for functions.
- ex thiscontainer (const exvector &v) const override

  Similar to duplicate(), but with a preset sequence.
- ex thiscontainer (exvector &&v) const override

  Similar to duplicate(), but with a preset sequence (which gets pilfered).

- ex conjugate () const override

  Implementation of ex::conjugate for functions.
- ex real_part () const override

  Implementation of ex::real_part for functions.
- ex imag_part () const override

  Implementation of ex::imag_part for functions.
- void archive (archive_node &n) const override

  Archive the object.
- void read_archive (const archive_node &n, lst &syms) override

  Construct object from archive_node.
- bool info (unsigned inf) const override

  Implementation of ex::info for functions.
- ex power (const ex &exp) const
- unsigned get_serial () const
- std::string get_name () const

  Return the print name of the function.

Public Member Functions inherited from GiNaC::container< std::vector >

- container (STLT const &s)
- container (STLT &&v)
- container (exvector::const_iterator b, exvector::const_iterator e)
- container (std::initializer_list< ex > il)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- container & prepend (const ex &b)

  Add element at front.
- container & append (const ex &b)

  Add element at back.
- container & remove_first ()

  Remove first element.
- container & remove_last ()

  Remove last element.
- container & remove_all ()

  Remove all elements.
- container & sort ()

  Sort elements.
- container & unique ()

  Remove adjacent duplicate elements.
- const_iterator begin () const
- const_iterator end () const
- const_reverse_iterator rbegin () const
- const_reverse_iterator rend () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

Return a vector containing the free indices of an expression.

- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for derivatives.
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_context &c, unsigned level) const
- void do_print_csrc (const print_csrc &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const

Protected Member Functions inherited from GiNaC::function

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for functions.
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- ex pderivative (unsigned diff_param) const
- ex expl_derivative (const symbol &s) const
- bool lookup_remember_table (ex &result) const
- void store_remember_table (ex const &result) const

Protected Member Functions inherited from GiNaC::container< std::vector >

- virtual void printseq (const print_context &c, char openbracket, char delim, char closebracket, unsigned this↩ _precedence, unsigned upper_precedence=0) const

  Print sequence of contained elements.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python (const print_python &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const
- STLT subschildren (const exmap &m, unsigned options=0) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Member Functions inherited from GiNaC::container_storage< C >

- container_storage ()
- container_storage (size_t n, const ex &e)
- container_storage (std::initializer_list< ex > il)
- template<class In>
  container_storage (In b, In e)
- void reserve (size_t)
- ~container_storage ()
- void reserve (size_t n)
- void reserve (std::vector< ex > &v, size_t n)

Protected Attributes

- paramset parameter_set

  Set of parameter numbers with respect to which to take the derivative.

Protected Attributes inherited from GiNaC::function

- unsigned serial

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

Protected Attributes inherited from GiNaC::container_storage< C >

- STLT seq

Additional Inherited Members

Public Types inherited from GiNaC::container< std::vector >

- typedef STLT::const_iterator const_iterator
- typedef STLT::const_reverse_iterator const_reverse_iterator

Static Public Member Functions inherited from GiNaC::function

- static unsigned register_new (function_options const &opt)
- static unsigned find_function (const std::string &name, unsigned nparams)

  Find serial number of function by name and number of parameters.
- static std::vector< function_options > get_registered_functions ()

Static Public Attributes inherited from GiNaC::function

- static unsigned current_serial = 0

  This can be used as a hook for external applications.

Protected Types inherited from GiNaC::container< std::vector >

- typedef container_storage< std::vector >::STLT STLT

Protected Types inherited from GiNaC::container_storage< C >

- typedef C< ex > STLT

Static Protected Member Functions inherited from GiNaC::function

- static std::vector< function_options > & registered_functions ()

Static Protected Member Functions inherited from GiNaC::container< std::vector >

- static unsigned get_default_flags ()
- static char get_open_delim ()
- static char get_close_delim ()

Static Protected Member Functions inherited from GiNaC::container_storage< C >

- static void reserve (STLT &, size_t)

8.62.1 Detailed Description

This class represents the (abstract) derivative of a symbolic function.

It is used to represent the derivatives of functions that do not have a derivative or series expansion procedure defined.

8.62.2 Constructor & Destructor Documentation

8.62.2.1 fderivative() [1/3]

```
GiNaC::fderivative::fderivative (
            unsigned ser,
            unsigned param,
            const exvector & args)
```

Construct derivative with respect to one parameter.

Parameters

ser	Serial number of function
param	Number of parameter with respect to which to take the derivative
args	Arguments of derivative function

References GiNaC::function::function(), and parameter_set.

Referenced by GiNaC::conjugate_expl_derivative(), derivative(), GiNaC::imag_part_expl_derivative(), is_equal_same_type(), match_same_type(), GiNaC::function::pderivative(), GiNaC::real_part_expl_derivative(), thiscontainer(), and thiscontainer().

8.62.2.2 fderivative() [2/3]

```
GiNaC::fderivative::fderivative (
             unsigned ser,
             const paramset & params,
             const exvector & args)
```

Construct derivative with respect to multiple parameters.

Parameters

ser	Serial number of function
params	Set of numbers of parameters with respect to which to take the derivative
args	Arguments of derivative function

References GiNaC::function::function(), and parameter_set.

8.62.2.3 fderivative() [3/3]

```
GiNaC::fderivative::fderivative (
             unsigned ser,
             const paramset & params,
             exvector && v)
```

References GiNaC::function::function(), and parameter_set.

8.62.3 Member Function Documentation

8.62.3.1 print()

```
void GiNaC::fderivative::print (
             const print_context & c,
             unsigned level = 0) const  [override], [virtual]
```

Output to stream.

This performs double dispatch on the dynamic type of ∗this and the dynamic type of the supplied print context.

Parameters

c	print context object that describes the output formatting
level	value that is used to identify the precedence or indentation level for placing parentheses and formatting

Reimplemented from GiNaC::basic.

References GiNaC::basic::print().

8.62.3.2 eval()

```
ex GiNaC::fderivative::eval () const  [override], [virtual]
```

Perform automatic non-interruptive term rewriting rules.

Reimplemented from GiNaC::basic.

References GiNaC::function::function(), GiNaC::basic::hold(), parameter_set, GiNaC::function::pderivative(), GiNaC::function::registered_functions(), GiNaC::container_storage< C >::seq, and GiNaC::function::serial.

8.62.3.3 series()

```
ex GiNaC::fderivative::series (
             const relational & r,
             int order,
             unsigned options = 0) const  [override], [virtual]
```

The series expansion of derivatives falls back to Taylor expansion.

See also

basic::series

Reimplemented from GiNaC::basic.

References GiNaC::basic::series().

8.62.3.4 thiscontainer() [1/2]

```
ex GiNaC::fderivative::thiscontainer (
             const exvector & v) const  [override], [virtual]
```

Similar to duplicate(), but with a preset sequence.

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References fderivative(), parameter_set, and GiNaC::function::serial.

8.62.3.5 thiscontainer() [2/2]

```
ex GiNaC::fderivative::thiscontainer (
             exvector && v) const  [override], [virtual]
```

Similar to duplicate(), but with a preset sequence (which gets pilfered).

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References fderivative(), parameter_set, and GiNaC::function::serial.

8.62.3.6 archive()

```
void GiNaC::fderivative::archive (
            archive_node & n) const  [override], [virtual]
```

Archive the object.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::archive_node::add_unsigned(), GiNaC::container< std::vector >::end(), and parameter_set.

8.62.3.7 read_archive()

```
void GiNaC::fderivative::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::archive_node::find_unsigned(), and parameter_set.

8.62.3.8 derivative()

```
ex GiNaC::fderivative::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for derivatives.

It applies the chain rule.

See also

ex::diff

Reimplemented from GiNaC::basic.

References fderivative(), GiNaC::ex::is_zero(), parameter_set, GiNaC::container_storage< C >::seq, and GiNaC::function::serial.

8.62.3.9 is_equal_same_type()

```
bool GiNaC::fderivative::is_equal_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if two objects of same type are equal.

Normally needs not be reimplemented as long as it wasn't overwritten by some parent class, since it just calls compare_same_type(). The reason why this function exists is that sometimes it is easier to determine equality than an order relation and then it can be overridden.

Reimplemented from GiNaC::container< std::vector >.

References fderivative(), GINAC_ASSERT, GiNaC::is_a(), and parameter_set.

8.62.3.10 match_same_type()

```
bool GiNaC::fderivative::match_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::basic.

References fderivative(), GINAC_ASSERT, GiNaC::is_a(), and parameter_set.

8.62.3.11 derivatives()

```
const paramset & GiNaC::fderivative::derivatives () const
```

Expose this object's derivative structure.

Parameter numbers occurring more than once stand for repeated differentiation with respect to that parameter. If a symbolic function f(x,y) is differentiated with respect to x, this method will return {0}. If f(x,y) is differentiated twice with respect to y, it will return {1,1}. (This corresponds to the way this object is printed.)

Returns

multiset of function's parameter numbers that are abstractly differentiated.

References parameter_set.

8.62.3.12 do_print()

```
void GiNaC::fderivative::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References GiNaC::container< std::vector >::end(), parameter_set, GiNaC::container< std::vector >::precedence(), GiNaC::function::precedence(), GiNaC::container< std::vector >::printseq(), GiNaC::function::registered_functions(), GiNaC::print_context::s, and GiNaC::function::serial.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.62.3.13 do_print_latex()

```
void GiNaC::fderivative::do_print_latex (
            const print_context & c,
            unsigned level) const  [protected]
```

References GiNaC::container< std::vector >::end(), parameter_set, GiNaC::container< std::vector >::precedence(), GiNaC::function::precedence(), GiNaC::container< std::vector >::printseq(), GiNaC::function::registered_functions(), GiNaC::print_context::s, and GiNaC::function::serial.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.62.3.14 do_print_csrc()

```
void GiNaC::fderivative::do_print_csrc (
            const print_csrc & c,
            unsigned level) const  [protected]
```

References GiNaC::container< std::vector >::end(), parameter_set, GiNaC::container< std::vector >::precedence(), GiNaC::function::precedence(), GiNaC::container< std::vector >::printseq(), GiNaC::function::registered_functions(), GiNaC::print_context::s, and GiNaC::function::serial.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.62.3.15 do_print_tree()

```
void GiNaC::fderivative::do_print_tree (
            const print_tree & c,
            unsigned level) const  [protected]
```

References GiNaC::print_tree::delta_indent, GiNaC::container< std::vector >::end(), GiNaC::basic::flags, GiNaC::basic::hashvalue, GiNaC::container< std::vector >::nops(), parameter_set, GiNaC::function::registered_functions(), GiNaC::print_context::s, GiNaC::container_storage< C >::seq, and GiNaC::function::serial.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.62.4 Member Data Documentation

8.62.4.1 parameter_set

```
paramset GiNaC::fderivative::parameter_set  [protected]
```

Set of parameter numbers with respect to which to take the derivative.

Referenced by archive(), derivative(), derivatives(), do_print(), do_print_csrc(), do_print_latex(), do_print_tree(), eval(), fderivative(), fderivative(), fderivative(), is_equal_same_type(), match_same_type(), read_archive(), thiscontainer(), and thiscontainer().

The documentation for this class was generated from the following files:

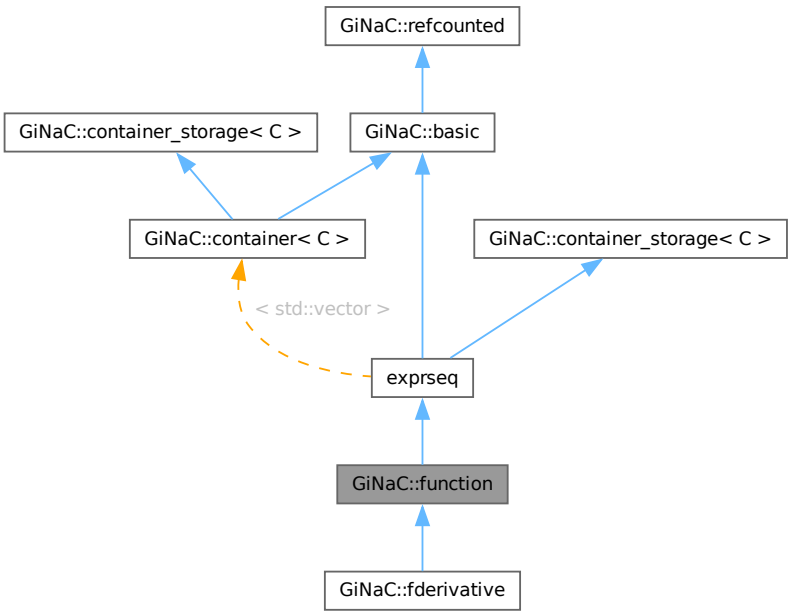
- fderivative.h
- fderivative.cpp

8.63 GiNaC::function Class Reference

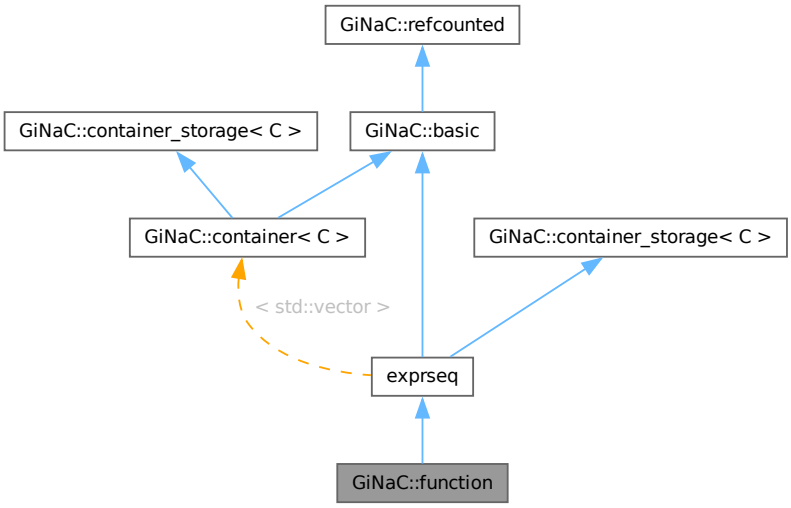
The class function is used to implement builtin functions like sin, cos... and user defined functions.

```
#include <function.h>
```

Inheritance diagram for GiNaC::function:



Collaboration diagram for GiNaC::function:



Public Member Functions

- function (unsigned ser)
- function (unsigned ser, const ex &param1)
- function (unsigned ser, const ex &param1, const ex &param2)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10, const ex &param11)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10, const ex &param11, const ex &param12)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10, const ex &param11, const ex &param12, const ex &param13)
- function (unsigned ser, const ex &param1, const ex &param2, const ex &param3, const ex &param4, const ex &param5, const ex &param6, const ex &param7, const ex &param8, const ex &param9, const ex &param10, const ex &param11, const ex &param12, const ex &param13, const ex &param14)
- function (unsigned ser, const exprseq &es)

- function (unsigned ser, const exvector &v)
- function (unsigned ser, exvector &&v)
- void print (const print_context &c, unsigned level=0) const override

  Output to stream.
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex evalf () const override

  Evaluate object numerically.
- ex eval_ncmul (const exvector &v) const override

  This method is defined to be in line with behavior of function::return_type().
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- ex series (const relational &r, int order, unsigned options=0) const override

  Implementation of ex::series for functions.
- ex thiscontainer (const exvector &v) const override

  Similar to duplicate(), but with a preset sequence.
- ex thiscontainer (exvector &&v) const override

  Similar to duplicate(), but with a preset sequence (which gets pilfered).
- ex conjugate () const override

  Implementation of ex::conjugate for functions.
- ex real_part () const override

  Implementation of ex::real_part for functions.
- ex imag_part () const override

  Implementation of ex::imag_part for functions.
- void archive (archive_node &n) const override

  Archive the object.
- void read_archive (const archive_node &n, lst &syms) override

  Construct object from archive_node.
- bool info (unsigned inf) const override

  Implementation of ex::info for functions.
- ex power (const ex &exp) const
- unsigned get_serial () const
- std::string get_name () const

  Return the print name of the function.

Public Member Functions inherited from GiNaC::container< std::vector >

- container (STLT const &s)
- container (STLT &&v)
- container (exvector::const_iterator b, exvector::const_iterator e)
- container (std::initializer_list< ex > il)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

Return modifiable operand/member at position i.

- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- container & prepend (const ex &b)

  Add element at front.
- container & append (const ex &b)

  Add element at back.
- container & remove_first ()

  Remove first element.
- container & remove_last ()

  Remove last element.
- container & remove_all ()

  Remove all elements.
- container & sort ()

  Sort elements.
- container & unique ()

  Remove adjacent duplicate elements.
- const_iterator begin () const
- const_iterator end () const
- const_reverse_iterator rbegin () const
- const_reverse_iterator rend () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Static Public Member Functions

- static unsigned register_new (function_options const &opt)
- static unsigned find_function (const std::string &name, unsigned nparams)

  Find serial number of function by name and number of parameters.
- static std::vector< function_options > get_registered_functions ()

Static Public Attributes

- static unsigned current_serial = 0

  This can be used as a hook for external applications.

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for functions.
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- ex pderivative (unsigned diff_param) const
- ex expl_derivative (const symbol &s) const
- bool lookup_remember_table (ex &result) const
- void store_remember_table (ex const &result) const

Protected Member Functions inherited from GiNaC::container< std::vector >

- virtual void printseq (const print_context &c, char openbracket, char delim, char closebracket, unsigned this↩
  _precedence, unsigned upper_precedence=0) const

  Print sequence of contained elements.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python (const print_python &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const
- STLT subschildren (const exmap &m, unsigned options=0) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Member Functions inherited from GiNaC::container_storage< C >

- container_storage ()
- container_storage (size_t n, const ex &e)
- container_storage (std::initializer_list< ex > il)
- template<class In>
  container_storage (In b, In e)
- void reserve (size_t)
- ~container_storage ()
- void reserve (size_t n)
- void reserve (std::vector< ex > &v, size_t n)

Static Protected Member Functions

- static std::vector< function_options > & registered_functions ()

Static Protected Member Functions inherited from GiNaC::container< std::vector >

- static unsigned get_default_flags ()
- static char get_open_delim ()
- static char get_close_delim ()

Static Protected Member Functions inherited from GiNaC::container_storage< C >

- static void reserve (STLT &, size_t)

Protected Attributes

- unsigned serial

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

Protected Attributes inherited from GiNaC::container_storage< C >

- STLT seq

Friends

- class remember_table_entry

Additional Inherited Members

Public Types inherited from GiNaC::container< std::vector >

- typedef STLT::const_iterator const_iterator
- typedef STLT::const_reverse_iterator const_reverse_iterator

Protected Types inherited from GiNaC::container< std::vector >

- typedef container_storage< std::vector >::STLT STLT

Protected Types inherited from GiNaC::container_storage< C >

- typedef C< ex > STLT

8.63.1 Detailed Description

The class function is used to implement builtin functions like sin, cos... and user defined functions.

8.63.2 Constructor & Destructor Documentation

8.63.2.1 function() [1/18]

```
GiNaC::function::function (
            unsigned ser)
```

References serial.

Referenced by GiNaC::fderivative::eval(), evalf(), GiNaC::fderivative::fderivative(), GiNaC::fderivative::fderivative(), GiNaC::fderivative::fderivative(), GiNaC::G(), GiNaC::G(), is_equal_same_type(), GiNaC::iterated_integral(), GiNaC::iterated_integral(), match_same_type(), GiNaC::psi(), GiNaC::psi(), remember_table_entry, thiscontainer(), thiscontainer(), GiNaC::zeta(), and GiNaC::zeta().

8.63.2.2 function() [2/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1)
```

References serial.

8.63.2.3 function() [3/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1,
            const ex & param2)
```

References serial.

8.63.2.4 function() [4/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1,
            const ex & param2,
            const ex & param3)
```

References serial.

8.63.2.5 function() [5/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1,
            const ex & param2,
            const ex & param3,
            const ex & param4)
```

References serial.

8.63.2.6 function() [6/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1,
            const ex & param2,
            const ex & param3,
            const ex & param4,
            const ex & param5)
```

References serial.

8.63.2.7 function() [7/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1,
            const ex & param2,
            const ex & param3,
            const ex & param4,
            const ex & param5,
            const ex & param6)
```

References serial.

8.63.2.8 function() [8/18]

```
GiNaC::function::function (
             unsigned ser,
             const ex & param1,
             const ex & param2,
             const ex & param3,
             const ex & param4,
             const ex & param5,
             const ex & param6,
             const ex & param7)
```

References serial.

8.63.2.9 function() [9/18]

```
GiNaC::function::function (
             unsigned ser,
             const ex & param1,
             const ex & param2,
             const ex & param3,
             const ex & param4,
             const ex & param5,
             const ex & param6,
             const ex & param7,
             const ex & param8)
```

References serial.

8.63.2.10 function() [10/18]

```
GiNaC::function::function (
             unsigned ser,
             const ex & param1,
             const ex & param2,
             const ex & param3,
             const ex & param4,
             const ex & param5,
             const ex & param6,
             const ex & param7,
             const ex & param8,
             const ex & param9)
```

References serial.

8.63.2.11 function() [11/18]

```
GiNaC::function::function (
             unsigned ser,
             const ex & param1,
             const ex & param2,
             const ex & param3,
             const ex & param4,
```

```
            const ex & param5,
            const ex & param6,
            const ex & param7,
            const ex & param8,
            const ex & param9,
            const ex & param10)
```

References serial.

8.63.2.12 function() [12/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1,
            const ex & param2,
            const ex & param3,
            const ex & param4,
            const ex & param5,
            const ex & param6,
            const ex & param7,
            const ex & param8,
            const ex & param9,
            const ex & param10,
            const ex & param11)
```

References serial.

8.63.2.13 function() [13/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1,
            const ex & param2,
            const ex & param3,
            const ex & param4,
            const ex & param5,
            const ex & param6,
            const ex & param7,
            const ex & param8,
            const ex & param9,
            const ex & param10,
            const ex & param11,
            const ex & param12)
```

References serial.

8.63.2.14 function() [14/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1,
            const ex & param2,
            const ex & param3,
```

```
            const ex & param4,
            const ex & param5,
            const ex & param6,
            const ex & param7,
            const ex & param8,
            const ex & param9,
            const ex & param10,
            const ex & param11,
            const ex & param12,
            const ex & param13)
```

References serial.

8.63.2.15 function() [15/18]

```
GiNaC::function::function (
            unsigned ser,
            const ex & param1,
            const ex & param2,
            const ex & param3,
            const ex & param4,
            const ex & param5,
            const ex & param6,
            const ex & param7,
            const ex & param8,
            const ex & param9,
            const ex & param10,
            const ex & param11,
            const ex & param12,
            const ex & param13,
            const ex & param14)
```

References serial.

8.63.2.16 function() [16/18]

```
GiNaC::function::function (
            unsigned ser,
            const exprseq & es)
```

References GiNaC::basic::clearflag(), GiNaC::status_flags::evaluated, and serial.

8.63.2.17 function() [17/18]

```
GiNaC::function::function (
            unsigned ser,
            const exvector & v)
```

References serial.

8.63.2.18 function() [18/18]

```
GiNaC::function::function (
             unsigned ser,
             exvector && v)
```

References serial.

8.63.3 Member Function Documentation

8.63.3.1 print()

```
void GiNaC::function::print (
             const print_context & c,
             unsigned level = 0) const  [override], [virtual]
```

Output to stream.

This performs double dispatch on the dynamic type of ∗this and the dynamic type of the supplied print context.

Parameters

c	print context object that describes the output formatting
level	value that is used to identify the precedence or indentation level for placing parentheses and formatting

Reimplemented from GiNaC::basic.

References current_serial, GiNaC::basic::flags, GiNaC::print_context_options::get_id(), GiNaC::class_info< OPT >::get_parent(), GINAC_ASSERT, GiNaC::basic::hashvalue, GiNaC::is_a(), GiNaC::function_options::name, GiNaC::container< std::vector >::nops() GiNaC::function_options::nparams, GiNaC::class_info< OPT >::options, GiNaC::container< std::vector >::precedence(), precedence(), print(), GiNaC::function_options::print_dispatch_table, GiNaC::function_options::print_use_exvector_args, GiNaC::container< std::vector >::printseq(), registered_functions(), GiNaC::print_context::s, GiNaC::container_storage< C >::seq, serial, and GiNaC::function_options::TeX_name.

Referenced by print(), and remember_table_entry.

8.63.3.2 precedence()

```
unsigned GiNaC::function::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::container< std::vector >.

References precedence().

Referenced by GiNaC::fderivative::do_print(), GiNaC::fderivative::do_print_csrc(), GiNaC::fderivative::do_print_latex(), precedence(), and print().

8.63.3.3 expand()

```
ex GiNaC::function::expand (
            unsigned options = 0) const  [override], [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented from GiNaC::basic.

References current_serial, GiNaC::function_options::expand_f, GiNaC::expand_options::expand_function_args, GiNaC::function_options::expand_use_exvector_args, GiNaC::status_flags::expanded, GINAC_ASSERT, GiNaC::function_options::n registered_functions(), GiNaC::container_storage< C >::seq, serial, and GiNaC::basic::setflag().

8.63.3.4 eval()

```
ex GiNaC::function::eval () const  [override], [virtual]
```

Perform automatic non-interruptive term rewriting rules.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::canonicalize(), current_serial, GiNaC::function_options::eval_f, GiNaC::function_options::eval_use_ GiNaC::status_flags::evaluated, GiNaC::ex::ex(), GiNaC::ex_to(), GiNaC::basic::flags, GINAC_ASSERT, GiNaC::basic::hold(), GiNaC::is_a(), GiNaC::ex::is_zero(), lookup_remember_table(), GiNaC::function_options::nparams, registered_functions(), GiNaC::container_storage< C >::seq, serial, store_remember_table(), GiNaC::function_options::symtree, thiscontainer(), and GiNaC::function_options::use_remember.

8.63.3.5 evalf()

```
ex GiNaC::function::evalf () const  [override], [virtual]
```

Evaluate object numerically.

Reimplemented from GiNaC::basic.

References current_serial, GiNaC::function_options::evalf_f, GiNaC::function_options::evalf_params_first, GiNaC::function_options::evalf_use_exvector_args, function(), GINAC_ASSERT, GiNaC::function_options::nparams, registered_functions(), GiNaC::container_storage< C >::seq, and serial.

Referenced by GiNaC::iterated_integral2_eval(), and GiNaC::iterated_integral3_eval().

8.63.3.6 eval_ncmul()

```
ex GiNaC::function::eval_ncmul (
            const exvector & v) const  [override], [virtual]
```

This method is defined to be in line with behavior of function::return_type().

Reimplemented from GiNaC::basic.

References GiNaC::container_storage< C >::seq.

8.63.3.7 calchash()

```
unsigned GiNaC::function::calchash () const  [override], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References GiNaC::status_flags::evaluated, GiNaC::basic::flags, GiNaC::ex::gethash(), GiNaC::golden_ratio_hash(), GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, GiNaC::container< std::vector >::nops(), GiNaC::container< std::vector >::op(), GiNaC::rotate_left(), serial, and GiNaC::basic::setflag().

8.63.3.8 series()

```
ex GiNaC::function::series (
            const relational & r,
            int order,
            unsigned options = 0) const  [override], [virtual]
```

Implementation of ex::series for functions.

@see ex::series

Reimplemented from GiNaC::basic.

References current_serial, GINAC_ASSERT, GiNaC::function_options::nparams, registered_functions(), GiNaC::container_storage< C >::seq, serial, GiNaC::basic::series(), GiNaC::function_options::series_f, and GiNaC::function_options::series_use_exvector_args.

8.63.3.9 thiscontainer() [1/2]

```
ex GiNaC::function::thiscontainer (
            const exvector & v) const  [override], [virtual]
```

Similar to duplicate(), but with a preset sequence.

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References function(), and serial.

Referenced by eval().

8.63.3.10 thiscontainer() [2/2]

```
ex GiNaC::function::thiscontainer (
            exvector && v) const  [override], [virtual]
```

Similar to duplicate(), but with a preset sequence (which gets pilfered).

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References function(), and serial.

8.63.3.11 conjugate()

```
ex GiNaC::function::conjugate () const  [override], [virtual]
```

Implementation of ex::conjugate for functions.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::function_options::conjugate_f, GiNaC::function_options::conjugate_use_exvector_args, GINAC_ASSERT, GiNaC::function_options::nparams, registered_functions(), GiNaC::container_storage< C >::seq, and serial.

8.63.3.12 real_part()

```
ex GiNaC::function::real_part () const  [override], [virtual]
```

Implementation of ex::real_part for functions.

Reimplemented from GiNaC::container< std::vector >.

References GINAC_ASSERT, GiNaC::function_options::nparams, GiNaC::basic::real_part(), GiNaC::function_options::real_part_f, GiNaC::function_options::real_part_use_exvector_args, registered_functions(), GiNaC::container_storage< C >::seq, and serial.

8.63.3.13 imag_part()

```
ex GiNaC::function::imag_part () const  [override], [virtual]
```

Implementation of ex::imag_part for functions.

Reimplemented from GiNaC::container< std::vector >.

References GINAC_ASSERT, GiNaC::basic::imag_part(), GiNaC::function_options::imag_part_f, GiNaC::function_options::imag_part
GiNaC::function_options::nparams, registered_functions(), GiNaC::container_storage< C >::seq, and serial.

8.63.3.14 archive()

```
void GiNaC::function::archive (
            archive_node & n) const  [override], [virtual]
```

Archive the object.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::archive_node::add_string(), GINAC_ASSERT, registered_functions(), and serial.

8.63.3.15 read_archive()

```
void GiNaC::function::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Construct object from archive_node.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::archive_node::find_string(), registered_functions(), GiNaC::container_storage< C >::seq, and serial.

8.63.3.16 info()

```
bool GiNaC::function::info (
            unsigned inf) const  [override], [virtual]
```

Implementation of ex::info for functions.

Reimplemented from GiNaC::container< std::vector >.

References GINAC_ASSERT, GiNaC::basic::info(), GiNaC::function_options::info_f, GiNaC::function_options::info_use_exvector_arg
GiNaC::function_options::nparams, registered_functions(), GiNaC::container_storage< C >::seq, and serial.

8.63.3.17 derivative()

```
ex GiNaC::function::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for functions.

It applies the chain rule, except for the Order term function. @see ex::diff

Reimplemented from GiNaC::basic.

References expl_derivative(), GiNaC::ex::is_zero(), pderivative(), and GiNaC::container_storage< C >::seq.

8.63.3.18 is_equal_same_type()

```
bool GiNaC::function::is_equal_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if two objects of same type are equal.

Normally needs not be reimplemented as long as it wasn't overwritten by some parent class, since it just calls compare_same_type(). The reason why this function exists is that sometimes it is easier to determine equality than an order relation and then it can be overridden.

Reimplemented from GiNaC::container< std::vector >.

References function(), GINAC_ASSERT, GiNaC::is_a(), GiNaC::container< std::vector >::is_equal_same_type(), and serial.

8.63.3.19 match_same_type()

```
bool GiNaC::function::match_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::basic.

References function(), GINAC_ASSERT, GiNaC::is_a(), and serial.

8.63.3.20 return_type()

```
unsigned GiNaC::function::return_type () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative, GINAC_ASSERT, registered_functions(), GiNaC::function_options::return_type, GiNaC::container_storage< C >::seq, serial, and GiNaC::function_options::use_return_type.

8.63.3.21 return_type_tinfo()

```
return_type_t GiNaC::function::return_type_tinfo () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GINAC_ASSERT, GiNaC::make_return_type_t(), registered_functions(), GiNaC::function_options::return_type_tinfo, GiNaC::container_storage< C >::seq, serial, and GiNaC::function_options::use_return_type.

8.63.3.22 pderivative()

```
ex GiNaC::function::pderivative (
            unsigned diff_param) const  [protected]
```

References current_serial, GiNaC::function_options::derivative_f, GiNaC::function_options::derivative_use_exvector_args, GiNaC::fderivative::fderivative(), GINAC_ASSERT, GiNaC::function_options::nparams, registered_functions(), GiNaC::container_storage< C >::seq, and serial.

Referenced by derivative(), and GiNaC::fderivative::eval().

8.63.3.23 expl_derivative()

```
ex GiNaC::function::expl_derivative (
            const symbol & s) const  [protected]
```

References current_serial, GiNaC::function_options::expl_derivative_f, GiNaC::function_options::expl_derivative_use_exvector_args, GINAC_ASSERT, GiNaC::function_options::nparams, registered_functions(), GiNaC::container_storage< C >::seq, and serial.

Referenced by derivative().

8.63.3.24 registered_functions()

```
std::vector< function_options > & GiNaC::function::registered_functions ()  [static], [protected]
```

Referenced by archive(), conjugate(), GiNaC::fderivative::do_print(), GiNaC::fderivative::do_print_csrc(), GiNaC::fderivative::do_print_latex(), GiNaC::fderivative::do_print_tree(), GiNaC::fderivative::eval(), eval(), evalf(), expand(), expl_derivative(), find_function(), get_name(), get_registered_functions(), imag_part(), info(), pderivative(), power(), print(), read_archive(), real_part(), register_new(), return_type(), return_type_tinfo(), and series().

8.63.3.25 lookup_remember_table()

```
bool GiNaC::function::lookup_remember_table (
            ex & result) const  [protected]
```

References GiNaC::remember_table::remember_tables(), and serial.

Referenced by eval().

8.63.3.26 store_remember_table()

```
void GiNaC::function::store_remember_table (
            ex const & result) const  [protected]
```

References GiNaC::remember_table::remember_tables(), and serial.

Referenced by eval().

8.63.3.27 power()

```
ex GiNaC::function::power (
            const ex & exp) const
```

References current_serial, GiNaC::dynallocate(), GiNaC::status_flags::evaluated, GINAC_ASSERT, GiNaC::function_options::nparam GiNaC::function_options::power_f, GiNaC::function_options::power_use_exvector_args, registered_functions(), GiNaC::container_storage< C >::seq, and serial.

8.63.3.28 register_new()

```
unsigned GiNaC::function::register_new (
            function_options const & opt)  [static]
```

References GiNaC::function_options::functions_with_same_name, GiNaC::function_options::name, registered_functions(), GiNaC::function_options::remember_assoc_size, GiNaC::function_options::remember_size, GiNaC::function_options::remember_str GiNaC::remember_table::remember_table(), GiNaC::remember_table::remember_tables(), and GiNaC::function_options::use_remem

8.63.3.29 find_function()

```
unsigned GiNaC::function::find_function (
            const std::string & name,
            unsigned nparams)  [static]
```

Find serial number of function by name and number of parameters.

Throws exception if function was not found.

References registered_functions(), and serial.

8.63.3.30 get_registered_functions()

```
std::vector< function_options > GiNaC::function::get_registered_functions ()  [inline], [static]
```

References registered_functions().

8.63.3.31 get_serial()

```
unsigned GiNaC::function::get_serial () const  [inline]
```

References serial.

8.63.3.32 get_name()

```
std::string GiNaC::function::get_name () const
```

Return the print name of the function.

References GINAC_ASSERT, registered_functions(), and serial.

8.63.4 Friends And Related Symbol Documentation

8.63.4.1 remember_table_entry

```
friend class remember_table_entry  [friend]
```

References function(), print(), and remember_table_entry.

Referenced by remember_table_entry.

8.63.5 Member Data Documentation

8.63.5.1 current_serial

```
unsigned GiNaC::function::current_serial = 0  [static]
```

This can be used as a hook for external applications.

Referenced by eval(), evalf(), expand(), expl_derivative(), pderivative(), power(), print(), and series().

8.63.5.2 serial

```
unsigned GiNaC::function::serial  [protected]
```

Referenced by archive(), calchash(), conjugate(), GiNaC::fderivative::derivative(), GiNaC::fderivative::do_print(), GiNaC::fderivative::do_print_csrc(), GiNaC::fderivative::do_print_latex(), GiNaC::fderivative::do_print_tree(), GiNaC::fderivative::eval(), eval(), evalf(), expand(), expl_derivative(), find_function(), function(), function(), function(), function(), function(), function(), function(), function(), function(), function(), function(), function(), function(), function(), function(), function(), function(), get_name(), get_serial(), imag_part(), info(), is_equal_same_type(), lookup_remember_table(), match_same_type(), pderivative(), power(), print(), read_archive(), real_part(), return_type(), return_type_tinfo(), series(), store_remember_table(), GiNaC::fderivative::thiscontainer(), GiNaC::fderivative::thiscontainer(), thiscontainer(), and thiscontainer().

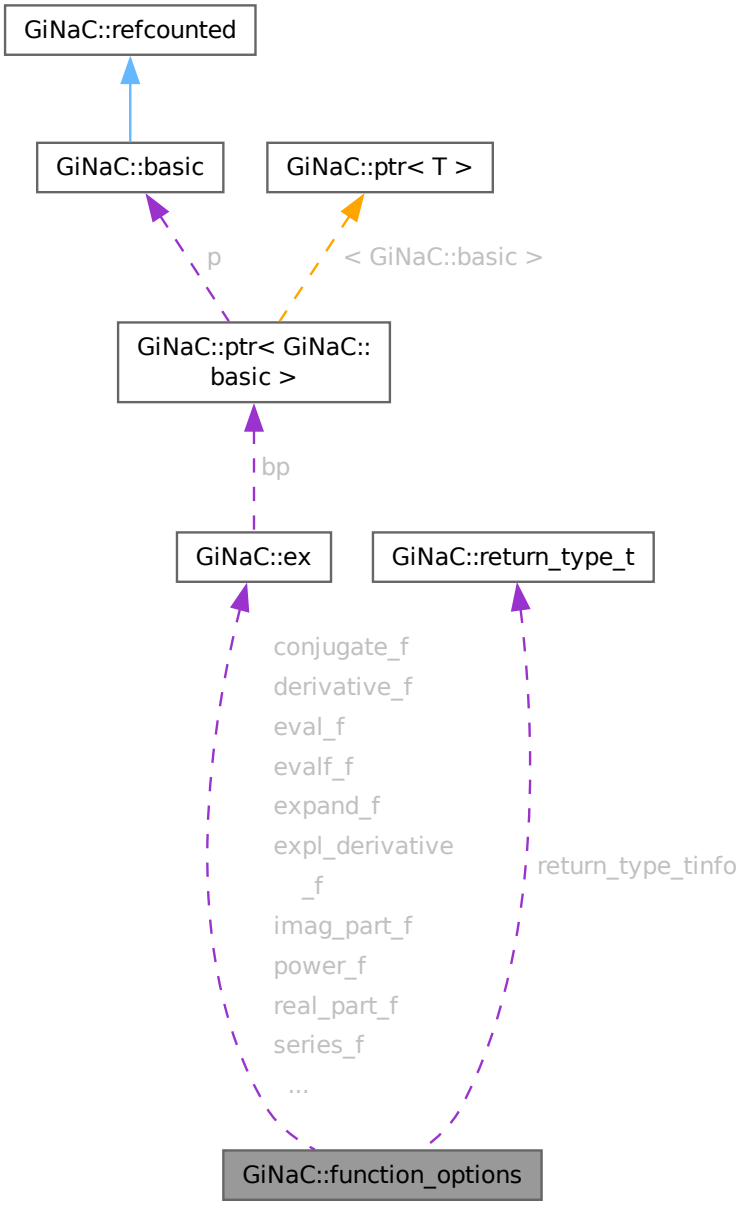
The documentation for this class was generated from the following files:

- function.h
- function.cpp

8.64 GiNaC::function_options Class Reference

`#include <function.h>`

Collaboration diagram for GiNaC::function_options:



Public Member Functions

- function_options ()
- function_options (std::string const &n, std::string const &tn=std::string())
- function_options (std::string const &n, unsigned np)

- ~function_options ()
- void initialize ()
- function_options & dummy ()
- function_options & set_name (std::string const &n, std::string const &tn=std::string())
- function_options & latex_name (std::string const &tn)
- function_options & eval_func (eval_funcp_1 e)
- function_options & eval_func (eval_funcp_2 e)
- function_options & eval_func (eval_funcp_3 e)
- function_options & eval_func (eval_funcp_4 e)
- function_options & eval_func (eval_funcp_5 e)
- function_options & eval_func (eval_funcp_6 e)
- function_options & eval_func (eval_funcp_7 e)
- function_options & eval_func (eval_funcp_8 e)
- function_options & eval_func (eval_funcp_9 e)
- function_options & eval_func (eval_funcp_10 e)
- function_options & eval_func (eval_funcp_11 e)
- function_options & eval_func (eval_funcp_12 e)
- function_options & eval_func (eval_funcp_13 e)
- function_options & eval_func (eval_funcp_14 e)
- function_options & evalf_func (evalf_funcp_1 e)
- function_options & evalf_func (evalf_funcp_2 e)
- function_options & evalf_func (evalf_funcp_3 e)
- function_options & evalf_func (evalf_funcp_4 e)
- function_options & evalf_func (evalf_funcp_5 e)
- function_options & evalf_func (evalf_funcp_6 e)
- function_options & evalf_func (evalf_funcp_7 e)
- function_options & evalf_func (evalf_funcp_8 e)
- function_options & evalf_func (evalf_funcp_9 e)
- function_options & evalf_func (evalf_funcp_10 e)
- function_options & evalf_func (evalf_funcp_11 e)
- function_options & evalf_func (evalf_funcp_12 e)
- function_options & evalf_func (evalf_funcp_13 e)
- function_options & evalf_func (evalf_funcp_14 e)
- function_options & conjugate_func (conjugate_funcp_1 e)
- function_options & conjugate_func (conjugate_funcp_2 e)
- function_options & conjugate_func (conjugate_funcp_3 e)
- function_options & conjugate_func (conjugate_funcp_4 e)
- function_options & conjugate_func (conjugate_funcp_5 e)
- function_options & conjugate_func (conjugate_funcp_6 e)
- function_options & conjugate_func (conjugate_funcp_7 e)
- function_options & conjugate_func (conjugate_funcp_8 e)
- function_options & conjugate_func (conjugate_funcp_9 e)
- function_options & conjugate_func (conjugate_funcp_10 e)
- function_options & conjugate_func (conjugate_funcp_11 e)
- function_options & conjugate_func (conjugate_funcp_12 e)
- function_options & conjugate_func (conjugate_funcp_13 e)
- function_options & conjugate_func (conjugate_funcp_14 e)
- function_options & real_part_func (real_part_funcp_1 e)
- function_options & real_part_func (real_part_funcp_2 e)
- function_options & real_part_func (real_part_funcp_3 e)
- function_options & real_part_func (real_part_funcp_4 e)
- function_options & real_part_func (real_part_funcp_5 e)
- function_options & real_part_func (real_part_funcp_6 e)
- function_options & real_part_func (real_part_funcp_7 e)
- function_options & real_part_func (real_part_funcp_8 e)

- function_options & real_part_func (real_part_funcp_9 e)
- function_options & real_part_func (real_part_funcp_10 e)
- function_options & real_part_func (real_part_funcp_11 e)
- function_options & real_part_func (real_part_funcp_12 e)
- function_options & real_part_func (real_part_funcp_13 e)
- function_options & real_part_func (real_part_funcp_14 e)
- function_options & imag_part_func (imag_part_funcp_1 e)
- function_options & imag_part_func (imag_part_funcp_2 e)
- function_options & imag_part_func (imag_part_funcp_3 e)
- function_options & imag_part_func (imag_part_funcp_4 e)
- function_options & imag_part_func (imag_part_funcp_5 e)
- function_options & imag_part_func (imag_part_funcp_6 e)
- function_options & imag_part_func (imag_part_funcp_7 e)
- function_options & imag_part_func (imag_part_funcp_8 e)
- function_options & imag_part_func (imag_part_funcp_9 e)
- function_options & imag_part_func (imag_part_funcp_10 e)
- function_options & imag_part_func (imag_part_funcp_11 e)
- function_options & imag_part_func (imag_part_funcp_12 e)
- function_options & imag_part_func (imag_part_funcp_13 e)
- function_options & imag_part_func (imag_part_funcp_14 e)
- function_options & expand_func (expand_funcp_1 e)
- function_options & expand_func (expand_funcp_2 e)
- function_options & expand_func (expand_funcp_3 e)
- function_options & expand_func (expand_funcp_4 e)
- function_options & expand_func (expand_funcp_5 e)
- function_options & expand_func (expand_funcp_6 e)
- function_options & expand_func (expand_funcp_7 e)
- function_options & expand_func (expand_funcp_8 e)
- function_options & expand_func (expand_funcp_9 e)
- function_options & expand_func (expand_funcp_10 e)
- function_options & expand_func (expand_funcp_11 e)
- function_options & expand_func (expand_funcp_12 e)
- function_options & expand_func (expand_funcp_13 e)
- function_options & expand_func (expand_funcp_14 e)
- function_options & derivative_func (derivative_funcp_1 e)
- function_options & derivative_func (derivative_funcp_2 e)
- function_options & derivative_func (derivative_funcp_3 e)
- function_options & derivative_func (derivative_funcp_4 e)
- function_options & derivative_func (derivative_funcp_5 e)
- function_options & derivative_func (derivative_funcp_6 e)
- function_options & derivative_func (derivative_funcp_7 e)
- function_options & derivative_func (derivative_funcp_8 e)
- function_options & derivative_func (derivative_funcp_9 e)
- function_options & derivative_func (derivative_funcp_10 e)
- function_options & derivative_func (derivative_funcp_11 e)
- function_options & derivative_func (derivative_funcp_12 e)
- function_options & derivative_func (derivative_funcp_13 e)
- function_options & derivative_func (derivative_funcp_14 e)
- function_options & expl_derivative_func (expl_derivative_funcp_1 e)
- function_options & expl_derivative_func (expl_derivative_funcp_2 e)
- function_options & expl_derivative_func (expl_derivative_funcp_3 e)
- function_options & expl_derivative_func (expl_derivative_funcp_4 e)
- function_options & expl_derivative_func (expl_derivative_funcp_5 e)
- function_options & expl_derivative_func (expl_derivative_funcp_6 e)
- function_options & expl_derivative_func (expl_derivative_funcp_7 e)

- function_options & expl_derivative_func (expl_derivative_funcp_8 e)
- function_options & expl_derivative_func (expl_derivative_funcp_9 e)
- function_options & expl_derivative_func (expl_derivative_funcp_10 e)
- function_options & expl_derivative_func (expl_derivative_funcp_11 e)
- function_options & expl_derivative_func (expl_derivative_funcp_12 e)
- function_options & expl_derivative_func (expl_derivative_funcp_13 e)
- function_options & expl_derivative_func (expl_derivative_funcp_14 e)
- function_options & power_func (power_funcp_1 e)
- function_options & power_func (power_funcp_2 e)
- function_options & power_func (power_funcp_3 e)
- function_options & power_func (power_funcp_4 e)
- function_options & power_func (power_funcp_5 e)
- function_options & power_func (power_funcp_6 e)
- function_options & power_func (power_funcp_7 e)
- function_options & power_func (power_funcp_8 e)
- function_options & power_func (power_funcp_9 e)
- function_options & power_func (power_funcp_10 e)
- function_options & power_func (power_funcp_11 e)
- function_options & power_func (power_funcp_12 e)
- function_options & power_func (power_funcp_13 e)
- function_options & power_func (power_funcp_14 e)
- function_options & series_func (series_funcp_1 e)
- function_options & series_func (series_funcp_2 e)
- function_options & series_func (series_funcp_3 e)
- function_options & series_func (series_funcp_4 e)
- function_options & series_func (series_funcp_5 e)
- function_options & series_func (series_funcp_6 e)
- function_options & series_func (series_funcp_7 e)
- function_options & series_func (series_funcp_8 e)
- function_options & series_func (series_funcp_9 e)
- function_options & series_func (series_funcp_10 e)
- function_options & series_func (series_funcp_11 e)
- function_options & series_func (series_funcp_12 e)
- function_options & series_func (series_funcp_13 e)
- function_options & series_func (series_funcp_14 e)
- function_options & info_func (info_funcp_1 e)
- function_options & info_func (info_funcp_2 e)
- function_options & info_func (info_funcp_3 e)
- function_options & info_func (info_funcp_4 e)
- function_options & info_func (info_funcp_5 e)
- function_options & info_func (info_funcp_6 e)
- function_options & info_func (info_funcp_7 e)
- function_options & info_func (info_funcp_8 e)
- function_options & info_func (info_funcp_9 e)
- function_options & info_func (info_funcp_10 e)
- function_options & info_func (info_funcp_11 e)
- function_options & info_func (info_funcp_12 e)
- function_options & info_func (info_funcp_13 e)
- function_options & info_func (info_funcp_14 e)
- function_options & eval_func (eval_funcp_exvector e)
- function_options & evalf_func (evalf_funcp_exvector e)
- function_options & conjugate_func (conjugate_funcp_exvector e)
- function_options & real_part_func (real_part_funcp_exvector e)
- function_options & imag_part_func (imag_part_funcp_exvector e)
- function_options & expand_func (expand_funcp_exvector e)

- function_options & derivative_func (derivative_funcp_exvector e)
- function_options & expl_derivative_func (expl_derivative_funcp_exvector e)
- function_options & power_func (power_funcp_exvector e)
- function_options & series_func (series_funcp_exvector e)
- function_options & info_func (info_funcp_exvector e)
- template<class Ctx>
  function_options & print_func (print_funcp_1 p)
- template<class Ctx>
  function_options & print_func (print_funcp_2 p)
- template<class Ctx>
  function_options & print_func (print_funcp_3 p)
- template<class Ctx>
  function_options & print_func (print_funcp_4 p)
- template<class Ctx>
  function_options & print_func (print_funcp_5 p)
- template<class Ctx>
  function_options & print_func (print_funcp_6 p)
- template<class Ctx>
  function_options & print_func (print_funcp_7 p)
- template<class Ctx>
  function_options & print_func (print_funcp_8 p)
- template<class Ctx>
  function_options & print_func (print_funcp_9 p)
- template<class Ctx>
  function_options & print_func (print_funcp_10 p)
- template<class Ctx>
  function_options & print_func (print_funcp_11 p)
- template<class Ctx>
  function_options & print_func (print_funcp_12 p)
- template<class Ctx>
  function_options & print_func (print_funcp_13 p)
- template<class Ctx>
  function_options & print_func (print_funcp_14 p)
- template<class Ctx>
  function_options & print_func (print_funcp_exvector p)
- function_options & set_return_type (unsigned rt, const return_type_t *rtt=nullptr)
- function_options & do_not_evalf_params ()
- function_options & remember (unsigned size, unsigned assoc_size=0, unsigned strategy=remember_strategies::delete_never)
- function_options & overloaded (unsigned o)
- function_options & set_symmetry (const symmetry &s)
- std::string get_name () const
- unsigned get_nparams () const

Protected Member Functions

- bool has_derivative () const
- bool has_power () const
- void test_and_set_nparams (unsigned n)
- void set_print_func (unsigned id, print_funcp f)

Protected Attributes

- std::string name
- std::string TeX_name
- unsigned nparams
- eval_funcp eval_f
- evalf_funcp evalf_f
- conjugate_funcp conjugate_f
- real_part_funcp real_part_f
- imag_part_funcp imag_part_f
- expand_funcp expand_f
- derivative_funcp derivative_f
- expl_derivative_funcp expl_derivative_f
- power_funcp power_f
- series_funcp series_f
- std::vector< print_funcp > print_dispatch_table
- info_funcp info_f
- bool evalf_params_first
- bool use_return_type
- unsigned return_type
- return_type_t return_type_tinfo
- bool use_remember
- unsigned remember_size
- unsigned remember_assoc_size
- unsigned remember_strategy
- bool eval_use_exvector_args
- bool evalf_use_exvector_args
- bool conjugate_use_exvector_args
- bool real_part_use_exvector_args
- bool imag_part_use_exvector_args
- bool expand_use_exvector_args
- bool derivative_use_exvector_args
- bool expl_derivative_use_exvector_args
- bool power_use_exvector_args
- bool series_use_exvector_args
- bool print_use_exvector_args
- bool info_use_exvector_args
- unsigned functions_with_same_name
- ex symtree

Friends

- class function
- class fderivative

8.64.1 Constructor & Destructor Documentation

8.64.1.1 function_options() [1/3]

```
GiNaC::function_options::function_options ()
```

References initialize().

Referenced by conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), conjugate_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), derivative_func(), do_not_evalf_params(), dummy(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), eval_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), evalf_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expand_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), expl_derivative_func(), fderivative, imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), imag_part_func(), info_func(), info_func(), info_func(), info_func(), info_func(), info_func(), info_func(), info_func(), info_func(), info_func(), info_func(), info_func(), info_func(), info_func(), latex_name(), overloaded(), power_func(), power_func(), power_func(), power_func(), power_func(), power_func(), power_func(), power_func(), power_func(), power_func(), power_func(), power_func(), power_func(), power_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), real_part_func(), remember(), series_func(), series_func(), series_func(), series_func(), series_func(), series_func(), series_func(), series_func(), series_func(), series_func(), series_func(), series_func(), series_func(), series_func(), set_name(), set_return_type(), and set_symmetry().

8.64.1.2 function_options() [2/3]

```
GiNaC::function_options::function_options (
            std::string const & n,
            std::string const & tn = std::string())
```

References initialize(), and set_name().

8.64.1.3 function_options() [3/3]

```
GiNaC::function_options::function_options (
            std::string const & n,
            unsigned np)
```

References initialize(), nparams, and set_name().

8.64.1.4 ~function_options()

```
GiNaC::function_options::~function_options ()
```

8.64.2 Member Function Documentation

8.64.2.1 initialize()

```
void GiNaC::function_options::initialize ()
```

References conjugate_f, conjugate_use_exvector_args, derivative_f, derivative_use_exvector_args, eval_f, eval_use_exvector_args, evalf_f, evalf_params_first, evalf_use_exvector_args, expand_f, expand_use_exvector_args, expl_derivative_f, expl_derivative_use_exvector_args, functions_with_same_name, imag_part_f, imag_part_use_exvector_args, info_f, info_use_exvector_args, nparams, power_f, power_use_exvector_args, print_use_exvector_args, real_part_f, real_part_use_exvector_args, series_f, series_use_exvector_args, set_name(), symtree, use_remember, and use_return_type.

Referenced by fderivative, function_options(), function_options(), and function_options().

8.64.2.2 dummy()

```
function_options & GiNaC::function_options::dummy ()  [inline]
```

References function_options().

8.64.2.3 set_name()

```
function_options & GiNaC::function_options::set_name (
            std::string const & n,
            std::string const & tn = std::string())
```

References function_options(), name, and TeX_name.

Referenced by function_options(), function_options(), and initialize().

8.64.2.4 latex_name()

```
function_options & GiNaC::function_options::latex_name (
            std::string const & tn)
```

References function_options(), and TeX_name.

8.64.2.5 eval_func() [1/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_1 e)
```

References eval_f, function_options(), and test_and_set_nparams().

8.64.2.6 eval_func() [2/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_2 e)
```

References function_options().

8.64.2.7 eval_func() [3/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_3 e)
```

References function_options().

8.64.2.8 eval_func() [4/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_4 e)
```

References function_options().

8.64.2.9 eval_func() [5/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_5 e)
```

References function_options().

8.64.2.10 eval_func() [6/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_6 e)
```

References function_options().

8.64.2.11 eval_func() [7/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_7 e)
```

References function_options().

8.64.2.12 eval_func() [8/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_8 e)
```

References function_options().

8.64.2.13 eval_func() [9/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_9 e)
```

References function_options().

8.64.2.14 eval_func() [10/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_10 e)
```

References function_options().

8.64.2.15 eval_func() [11/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_11 e)
```

References function_options().

8.64.2.16 eval_func() [12/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_12 e)
```

References function_options().

8.64.2.17 eval_func() [13/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_13 e)
```

References function_options().

8.64.2.18 eval_func() [14/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_14 e)
```

References evalf_func(), and function_options().

8.64.2.19 evalf_func() [1/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_1 e)
```

References evalf_f, function_options(), and test_and_set_nparams().

Referenced by eval_func().

8.64.2.20 evalf_func() [2/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_2 e)
```

References function_options().

8.64.2.21 evalf_func() [3/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_3 e)
```

References function_options().

8.64.2.22 evalf_func() [4/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_4 e)
```

References function_options().

8.64.2.23 evalf_func() [5/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_5 e)
```

References function_options().

8.64.2.24 evalf_func() [6/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_6 e)
```

References function_options().

8.64.2.25 evalf_func() [7/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_7 e)
```

References function_options().

8.64.2.26 evalf_func() [8/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_8 e)
```

References function_options().

8.64.2.27 evalf_func() [9/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_9 e)
```

References function_options().

8.64.2.28 evalf_func() [10/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_10 e)
```

References function_options().

8.64.2.29 evalf_func() [11/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_11 e)
```

References function_options().

8.64.2.30 evalf_func() [12/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_12 e)
```

References function_options().

8.64.2.31 evalf_func() [13/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_13 e)
```

References function_options().

8.64.2.32 evalf_func() [14/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_14 e)
```

References conjugate_func(), and function_options().

8.64.2.33 conjugate_func() [1/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_1 e)
```

References conjugate_f, function_options(), and test_and_set_nparams().

Referenced by evalf_func().

8.64.2.34 conjugate_func() [2/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_2 e)
```

References function_options().

8.64.2.35 conjugate_func() [3/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_3 e)
```

References function_options().

8.64.2.36 conjugate_func() [4/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_4 e)
```

References function_options().

8.64.2.37 conjugate_func() [5/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_5 e)
```

References function_options().

8.64.2.38 conjugate_func() [6/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_6 e)
```

References function_options().

8.64.2.39 conjugate_func() [7/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_7 e)
```

References function_options().

8.64.2.40 conjugate_func() [8/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_8 e)
```

References function_options().

8.64.2.41 conjugate_func() [9/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_9 e)
```

References function_options().

8.64.2.42 conjugate_func() [10/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_10 e)
```

References function_options().

8.64.2.43 conjugate_func() [11/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_11 e)
```

References function_options().

8.64.2.44 conjugate_func() [12/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_12 e)
```

References function_options().

8.64.2.45 conjugate_func() [13/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_13 e)
```

References function_options().

8.64.2.46 conjugate_func() [14/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_14 e)
```

References function_options(), and real_part_func().

8.64.2.47 real_part_func() [1/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_1 e)
```

References function_options(), real_part_f, and test_and_set_nparams().

Referenced by conjugate_func().

8.64.2.48 real_part_func() [2/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_2 e)
```

References function_options().

8.64.2.49 real_part_func() [3/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_3 e)
```

References function_options().

8.64.2.50 real_part_func() [4/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_4 e)
```

References function_options().

8.64.2.51 real_part_func() [5/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_5 e)
```

References function_options().

8.64.2.52 real_part_func() [6/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_6 e)
```

References function_options().

8.64.2.53 real_part_func() [7/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_7 e)
```

References function_options().

8.64.2.54 real_part_func() [8/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_8 e)
```

References function_options().

8.64.2.55 real_part_func() [9/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_9 e)
```

References function_options().

8.64.2.56 real_part_func() [10/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_10 e)
```

References function_options().

8.64.2.57 real_part_func() [11/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_11 e)
```

References function_options().

8.64.2.58 real_part_func() [12/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_12 e)
```

References function_options().

8.64.2.59 real_part_func() [13/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_13 e)
```

References function_options().

8.64.2.60 real_part_func() [14/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_14 e)
```

References function_options(), and imag_part_func().

8.64.2.61 imag_part_func() [1/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_1 e)
```

References function_options(), imag_part_f, and test_and_set_nparams().

Referenced by real_part_func().

8.64.2.62 imag_part_func() [2/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_2 e)
```

References function_options().

8.64.2.63 imag_part_func() [3/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_3 e)
```

References function_options().

8.64.2.64 imag_part_func() [4/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_4 e)
```

References function_options().

8.64.2.65 imag_part_func() [5/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_5 e)
```

References function_options().

8.64.2.66 imag_part_func() [6/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_6 e)
```

References function_options().

8.64.2.67 imag_part_func() [7/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_7 e)
```

References function_options().

8.64.2.68 imag_part_func() [8/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_8 e)
```

References function_options().

8.64.2.69 imag_part_func() [9/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_9 e)
```

References function_options().

8.64.2.70 imag_part_func() [10/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_10 e)
```

References function_options().

8.64.2.71 imag_part_func() [11/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_11 e)
```

References function_options().

8.64.2.72 imag_part_func() [12/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_12 e)
```

References function_options().

8.64.2.73 imag_part_func() [13/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_13 e)
```

References function_options().

8.64.2.74 imag_part_func() [14/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_14 e)
```

References expand_func(), and function_options().

8.64.2.75 expand_func() [1/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_1 e)
```

References expand_f, function_options(), and test_and_set_nparams().

Referenced by imag_part_func().

8.64.2.76 expand_func() [2/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_2 e)
```

References function_options().

8.64.2.77 expand_func() [3/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_3 e)
```

References function_options().

8.64.2.78 expand_func() [4/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_4 e)
```

References function_options().

8.64.2.79 expand_func() [5/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_5 e)
```

References function_options().

8.64.2.80 expand_func() [6/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_6 e)
```

References function_options().

8.64.2.81 expand_func() [7/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_7 e)
```

References function_options().

8.64.2.82 expand_func() [8/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_8 e)
```

References function_options().

8.64.2.83 expand_func() [9/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_9 e)
```

References function_options().

8.64.2.84 expand_func() [10/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_10 e)
```

References function_options().

8.64.2.85 expand_func() [11/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_11 e)
```

References function_options().

8.64.2.86 expand_func() [12/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_12 e)
```

References function_options().

8.64.2.87 expand_func() [13/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_13 e)
```

References function_options().

8.64.2.88 expand_func() [14/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_14 e)
```

References derivative_func(), and function_options().

8.64.2.89 derivative_func() [1/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_1 e)
```

References derivative_f, function_options(), and test_and_set_nparams().

Referenced by expand_func().

8.64.2.90 derivative_func() [2/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_2 e)
```

References function_options().

8.64.2.91 derivative_func() [3/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_3 e)
```

References function_options().

8.64.2.92 derivative_func() [4/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_4 e)
```

References function_options().

8.64.2.93 derivative_func() [5/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_5 e)
```

References function_options().

8.64.2.94 derivative_func() [6/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_6 e)
```

References function_options().

8.64.2.95 derivative_func() [7/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_7 e)
```

References function_options().

8.64.2.96 derivative_func() [8/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_8 e)
```

References function_options().

8.64.2.97 derivative_func() [9/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_9 e)
```

References function_options().

8.64.2.98 derivative_func() [10/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_10 e)
```

References function_options().

8.64.2.99 derivative_func() [11/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_11 e)
```

References function_options().

8.64.2.100 derivative_func() [12/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_12 e)
```

References function_options().

8.64.2.101 derivative_func() [13/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_13 e)
```

References function_options().

8.64.2.102 derivative_func() [14/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_14 e)
```

References expl_derivative_func(), and function_options().

8.64.2.103 expl_derivative_func() [1/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_1 e)
```

References expl_derivative_f, function_options(), and test_and_set_nparams().

Referenced by derivative_func().

8.64.2.104 expl_derivative_func() [2/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_2 e)
```

References function_options().

8.64.2.105 expl_derivative_func() [3/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_3 e)
```

References function_options().

8.64.2.106 expl_derivative_func() [4/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_4 e)
```

References function_options().

8.64.2.107 expl_derivative_func() [5/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_5 e)
```

References function_options().

8.64.2.108 expl_derivative_func() [6/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_6 e)
```

References function_options().

8.64.2.109 expl_derivative_func() [7/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_7 e)
```

References function_options().

8.64.2.110 expl_derivative_func() [8/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_8 e)
```

References function_options().

8.64.2.111 expl_derivative_func() [9/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_9 e)
```

References function_options().

8.64.2.112 expl_derivative_func() [10/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_10 e)
```

References function_options().

8.64.2.113 expl_derivative_func() [11/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_11 e)
```

References function_options().

8.64.2.114 expl_derivative_func() [12/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_12 e)
```

References function_options().

8.64.2.115 expl_derivative_func() [13/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_13 e)
```

References function_options().

8.64.2.116 expl_derivative_func() [14/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_14 e)
```

References function_options(), and power_func().

8.64.2.117 power_func() [1/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_1 e)
```

References function_options(), power_f, and test_and_set_nparams().

Referenced by expl_derivative_func().

8.64.2.118 power_func() [2/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_2 e)
```

References function_options().

8.64.2.119 power_func() [3/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_3 e)
```

References function_options().

8.64.2.120 power_func() [4/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_4 e)
```

References function_options().

8.64.2.121 power_func() [5/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_5 e)
```

References function_options().

8.64.2.122 power_func() [6/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_6 e)
```

References function_options().

8.64.2.123 power_func() [7/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_7 e)
```

References function_options().

8.64.2.124 power_func() [8/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_8 e)
```

References function_options().

8.64.2.125 power_func() [9/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_9 e)
```

References function_options().

8.64.2.126 power_func() [10/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_10 e)
```

References function_options().

8.64.2.127 power_func() [11/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_11 e)
```

References function_options().

8.64.2.128 power_func() [12/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_12 e)
```

References function_options().

8.64.2.129 power_func() [13/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_13 e)
```

References function_options().

8.64.2.130 power_func() [14/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_14 e)
```

References function_options(), and series_func().

8.64.2.131 series_func() [1/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_1 e)
```

References function_options(), series_f, and test_and_set_nparams().

Referenced by power_func().

8.64.2.132 series_func() [2/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_2 e)
```

References function_options().

8.64.2.133 series_func() [3/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_3 e)
```

References function_options().

8.64.2.134 series_func() [4/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_4 e)
```

References function_options().

8.64.2.135 series_func() [5/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_5 e)
```

References function_options().

8.64.2.136 series_func() [6/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_6 e)
```

References function_options().

8.64.2.137 series_func() [7/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_7 e)
```

References function_options().

8.64.2.138 series_func() [8/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_8 e)
```

References function_options().

8.64.2.139 series_func() [9/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_9 e)
```

References function_options().

8.64.2.140 series_func() [10/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_10 e)
```

References function_options().

8.64.2.141 series_func() [11/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_11 e)
```

References function_options().

8.64.2.142 series_func() [12/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_12 e)
```

References function_options().

8.64.2.143 series_func() [13/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_13 e)
```

References function_options().

8.64.2.144 series_func() [14/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_14 e)
```

References function_options(), and info_func().

8.64.2.145 info_func() [1/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_1 e)
```

References function_options(), info_f, and test_and_set_nparams().

Referenced by series_func().

8.64.2.146 info_func() [2/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_2 e)
```

References function_options().

8.64.2.147 info_func() [3/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_3 e)
```

References function_options().

8.64.2.148 info_func() [4/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_4 e)
```

References function_options().

8.64.2.149 info_func() [5/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_5 e)
```

References function_options().

8.64.2.150 info_func() [6/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_6 e)
```

References function_options().

8.64.2.151 info_func() [7/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_7 e)
```

References function_options().

8.64.2.152 info_func() [8/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_8 e)
```

References function_options().

8.64.2.153 info_func() [9/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_9 e)
```

References function_options().

8.64.2.154 info_func() [10/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_10 e)
```

References function_options().

8.64.2.155 info_func() [11/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_11 e)
```

References function_options().

8.64.2.156 info_func() [12/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_12 e)
```

References function_options().

8.64.2.157 info_func() [13/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_13 e)
```

References function_options().

8.64.2.158 info_func() [14/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_14 e)
```

References function_options().

8.64.2.159 eval_func() [15/15]

```
function_options & GiNaC::function_options::eval_func (
            eval_funcp_exvector e)
```

References function_options().

8.64.2.160 evalf_func() [15/15]

```
function_options & GiNaC::function_options::evalf_func (
            evalf_funcp_exvector e)
```

References function_options().

8.64.2.161 conjugate_func() [15/15]

```
function_options & GiNaC::function_options::conjugate_func (
            conjugate_funcp_exvector e)
```

References function_options().

8.64.2.162 real_part_func() [15/15]

```
function_options & GiNaC::function_options::real_part_func (
            real_part_funcp_exvector e)
```

References function_options().

8.64.2.163 imag_part_func() [15/15]

```
function_options & GiNaC::function_options::imag_part_func (
            imag_part_funcp_exvector e)
```

References function_options().

8.64.2.164 expand_func() [15/15]

```
function_options & GiNaC::function_options::expand_func (
            expand_funcp_exvector e)
```

References function_options().

8.64.2.165 derivative_func() [15/15]

```
function_options & GiNaC::function_options::derivative_func (
            derivative_funcp_exvector e)
```

References function_options().

8.64.2.166 expl_derivative_func() [15/15]

```
function_options & GiNaC::function_options::expl_derivative_func (
            expl_derivative_funcp_exvector e)
```

References function_options().

8.64.2.167 power_func() [15/15]

```
function_options & GiNaC::function_options::power_func (
            power_funcp_exvector e)
```

References function_options().

8.64.2.168 series_func() [15/15]

```
function_options & GiNaC::function_options::series_func (
            series_funcp_exvector e)
```

References function_options().

8.64.2.169 info_func() [15/15]

```
function_options & GiNaC::function_options::info_func (
            info_funcp_exvector e)
```

References function_options().

8.64.2.170 print_func() [1/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_1 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.171 print_func() [2/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_2 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.172 print_func() [3/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_3 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.173 print_func() [4/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_4 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.174 print_func() [5/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_5 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.175 print_func() [6/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_6 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.176 print_func() [7/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_7 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.177 print_func() [8/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_8 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.178 print_func() [9/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_9 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.179 print_func() [10/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_10 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.180 print_func() [11/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_11 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.181 print_func() [12/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_12 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.182 print_func() [13/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_13 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.183 print_func() [14/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_14 p)  [inline]
```

References function_options(), set_print_func(), and test_and_set_nparams().

8.64.2.184 print_func() [15/15]

```
template<class Ctx>
function_options & GiNaC::function_options::print_func (
            print_funcp_exvector p)  [inline]
```

References function_options(), print_use_exvector_args, and set_print_func().

8.64.2.185 set_return_type()

```
function_options & GiNaC::function_options::set_return_type (
            unsigned rt,
            const return_type_t * rtt = nullptr)
```

References function_options(), GiNaC::make_return_type_t(), return_type, return_type_tinfo, and use_return_type.

8.64.2.186 do_not_evalf_params()

```
function_options & GiNaC::function_options::do_not_evalf_params ()
```

References evalf_params_first, and function_options().

8.64.2.187 remember()

```
function_options & GiNaC::function_options::remember (
            unsigned size,
            unsigned assoc_size = 0,
            unsigned strategy = remember_strategies::delete_never)
```

References function_options(), remember_assoc_size, remember_size, remember_strategy, and use_remember.

8.64.2.188 overloaded()

```
function_options & GiNaC::function_options::overloaded (
            unsigned o)
```

References function_options(), and functions_with_same_name.

8.64.2.189 set_symmetry()

```
function_options & GiNaC::function_options::set_symmetry (
            const symmetry & s)
```

References function_options(), and symtree.

8.64.2.190 get_name()

```
std::string GiNaC::function_options::get_name () const  [inline]
```

References name.

8.64.2.191 get_nparams()

```
unsigned GiNaC::function_options::get_nparams () const  [inline]
```

References nparams.

8.64.2.192 has_derivative()

```
bool GiNaC::function_options::has_derivative () const  [inline], [protected]
```

References derivative_f.

8.64.2.193 has_power()

```
bool GiNaC::function_options::has_power () const  [inline], [protected]
```

References power_f.

8.64.2.194 test_and_set_nparams()

```
void GiNaC::function_options::test_and_set_nparams (
            unsigned n)  [protected]
```

References name, and nparams.

Referenced by conjugate_func(), derivative_func(), eval_func(), evalf_func(), expand_func(), expl_derivative_func(), imag_part_func(), info_func(), power_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), real_part_func(), and series_func().

8.64.2.195 set_print_func()

```
void GiNaC::function_options::set_print_func (
            unsigned id,
            print_funcp f)  [protected]
```

References print_dispatch_table.

Referenced by print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), print_func(), and print_func().

8.64.3 Friends And Related Symbol Documentation

8.64.3.1 function

```
friend class function  [friend]
```

References function.

Referenced by function.

8.64.3.2 fderivative

```
friend class fderivative  [friend]
```

References fderivative, function_options(), and initialize().

Referenced by fderivative.

8.64.4 Member Data Documentation

8.64.4.1 name

```
std::string GiNaC::function_options::name  [protected]
```

Referenced by get_name(), GiNaC::function::print(), GiNaC::function::register_new(), set_name(), and test_and_set_nparams().

8.64.4.2 TeX_name

```
std::string GiNaC::function_options::TeX_name  [protected]
```

Referenced by latex_name(), GiNaC::function::print(), and set_name().

8.64.4.3 nparams

```
unsigned GiNaC::function_options::nparams  [protected]
```

Referenced by GiNaC::function::conjugate(), GiNaC::function::eval(), GiNaC::function::evalf(), GiNaC::function::expand(), GiNaC::function::expl_derivative(), function_options(), get_nparams(), GiNaC::function::imag_part(), GiNaC::function::info(), initialize(), GiNaC::function::pderivative(), GiNaC::function::power(), GiNaC::function::print(), GiNaC::function::real_part(), GiNaC::function::series(), and test_and_set_nparams().

8.64.4.4 eval_f

```
eval_funcp GiNaC::function_options::eval_f  [protected]
```

Referenced by GiNaC::function::eval(), eval_func(), and initialize().

8.64.4.5 evalf_f

```
evalf_funcp GiNaC::function_options::evalf_f  [protected]
```

Referenced by GiNaC::function::evalf(), evalf_func(), and initialize().

8.64.4.6 conjugate_f

```
conjugate_funcp GiNaC::function_options::conjugate_f  [protected]
```

Referenced by GiNaC::function::conjugate(), conjugate_func(), and initialize().

8.64.4.7 real_part_f

real_part_funcp GiNaC::function_options::real_part_f [protected]

Referenced by initialize(), GiNaC::function::real_part(), and real_part_func().

8.64.4.8 imag_part_f

imag_part_funcp GiNaC::function_options::imag_part_f [protected]

Referenced by GiNaC::function::imag_part(), imag_part_func(), and initialize().

8.64.4.9 expand_f

expand_funcp GiNaC::function_options::expand_f [protected]

Referenced by GiNaC::function::expand(), expand_func(), and initialize().

8.64.4.10 derivative_f

derivative_funcp GiNaC::function_options::derivative_f [protected]

Referenced by derivative_func(), has_derivative(), initialize(), and GiNaC::function::pderivative().

8.64.4.11 expl_derivative_f

expl_derivative_funcp GiNaC::function_options::expl_derivative_f [protected]

Referenced by GiNaC::function::expl_derivative(), expl_derivative_func(), and initialize().

8.64.4.12 power_f

power_funcp GiNaC::function_options::power_f [protected]

Referenced by has_power(), initialize(), GiNaC::function::power(), and power_func().

8.64.4.13 series_f

series_funcp GiNaC::function_options::series_f [protected]

Referenced by initialize(), GiNaC::function::series(), and series_func().

8.64.4.14 print_dispatch_table

std::vector<print_funcp> GiNaC::function_options::print_dispatch_table [protected]

Referenced by GiNaC::function::print(), and set_print_func().

8.64.4.15 info_f

```
info_funcp GiNaC::function_options::info_f  [protected]
```

Referenced by GiNaC::function::info(), info_func(), and initialize().

8.64.4.16 evalf_params_first

```
bool GiNaC::function_options::evalf_params_first  [protected]
```

Referenced by do_not_evalf_params(), GiNaC::function::evalf(), and initialize().

8.64.4.17 use_return_type

```
bool GiNaC::function_options::use_return_type  [protected]
```

Referenced by initialize(), GiNaC::function::return_type(), GiNaC::function::return_type_tinfo(), and set_return_type().

8.64.4.18 return_type

```
unsigned GiNaC::function_options::return_type  [protected]
```

Referenced by GiNaC::function::return_type(), and set_return_type().

8.64.4.19 return_type_tinfo

```
return_type_t GiNaC::function_options::return_type_tinfo  [protected]
```

Referenced by GiNaC::function::return_type_tinfo(), and set_return_type().

8.64.4.20 use_remember

```
bool GiNaC::function_options::use_remember  [protected]
```

Referenced by GiNaC::function::eval(), initialize(), GiNaC::function::register_new(), and remember().

8.64.4.21 remember_size

```
unsigned GiNaC::function_options::remember_size  [protected]
```

Referenced by GiNaC::function::register_new(), and remember().

8.64.4.22 remember_assoc_size

```
unsigned GiNaC::function_options::remember_assoc_size  [protected]
```

Referenced by GiNaC::function::register_new(), and remember().

8.64.4.23 remember_strategy

```
unsigned GiNaC::function_options::remember_strategy  [protected]
```

Referenced by GiNaC::function::register_new(), and remember().

8.64.4.24 eval_use_exvector_args

```
bool GiNaC::function_options::eval_use_exvector_args  [protected]
```

Referenced by GiNaC::function::eval(), and initialize().

8.64.4.25 evalf_use_exvector_args

```
bool GiNaC::function_options::evalf_use_exvector_args  [protected]
```

Referenced by GiNaC::function::evalf(), and initialize().

8.64.4.26 conjugate_use_exvector_args

```
bool GiNaC::function_options::conjugate_use_exvector_args  [protected]
```

Referenced by GiNaC::function::conjugate(), and initialize().

8.64.4.27 real_part_use_exvector_args

```
bool GiNaC::function_options::real_part_use_exvector_args  [protected]
```

Referenced by initialize(), and GiNaC::function::real_part().

8.64.4.28 imag_part_use_exvector_args

```
bool GiNaC::function_options::imag_part_use_exvector_args  [protected]
```

Referenced by GiNaC::function::imag_part(), and initialize().

8.64.4.29 expand_use_exvector_args

```
bool GiNaC::function_options::expand_use_exvector_args  [protected]
```

Referenced by GiNaC::function::expand(), and initialize().

8.64.4.30 derivative_use_exvector_args

```
bool GiNaC::function_options::derivative_use_exvector_args  [protected]
```

Referenced by initialize(), and GiNaC::function::pderivative().

8.64.4.31 expl_derivative_use_exvector_args

```
bool GiNaC::function_options::expl_derivative_use_exvector_args  [protected]
```

Referenced by GiNaC::function::expl_derivative(), and initialize().

8.64.4.32 power_use_exvector_args

```
bool GiNaC::function_options::power_use_exvector_args  [protected]
```

Referenced by initialize(), and GiNaC::function::power().

8.64.4.33 series_use_exvector_args

```
bool GiNaC::function_options::series_use_exvector_args  [protected]
```

Referenced by initialize(), and GiNaC::function::series().

8.64.4.34 print_use_exvector_args

```
bool GiNaC::function_options::print_use_exvector_args  [protected]
```

Referenced by initialize(), GiNaC::function::print(), and print_func().

8.64.4.35 info_use_exvector_args

```
bool GiNaC::function_options::info_use_exvector_args  [protected]
```

Referenced by GiNaC::function::info(), and initialize().

8.64.4.36 functions_with_same_name

```
unsigned GiNaC::function_options::functions_with_same_name  [protected]
```

Referenced by initialize(), overloaded(), and GiNaC::function::register_new().

8.64.4.37 symtree

```
ex GiNaC::function_options::symtree  [protected]
```

Referenced by GiNaC::function::eval(), initialize(), and set_symmetry().

The documentation for this class was generated from the following files:

- function.h
- function.cpp

8.65 GiNaC::G2_SERIAL Class Reference

Generalized multiple polylogarithm.

`#include <inifcns.h>`

Static Public Attributes

- static unsigned serial

8.65.1 Detailed Description

Generalized multiple polylogarithm.

8.65.2 Member Data Documentation

8.65.2.1 serial

```
unsigned GiNaC::G2_SERIAL::serial  [static]
```

Initial value:

```
= function::register_new(function_options("G", 2).
     evalf_func(G2_evalf).
     eval_func(G2_eval).
     overloaded(2))
```

Referenced by GiNaC::G().

The documentation for this class was generated from the following files:

- inifcns.h
- inifcns_nstdsums.cpp

8.66 GiNaC::G3_SERIAL Class Reference

Generalized multiple polylogarithm with explicit imaginary parts.

`#include <inifcns.h>`

Static Public Attributes

- static unsigned serial

8.66.1 Detailed Description

Generalized multiple polylogarithm with explicit imaginary parts.

8.66.2 Member Data Documentation

8.66.2.1 serial

```
unsigned GiNaC::G3_SERIAL::serial  [static]
```

Initial value:

```
= function::register_new(function_options("G", 3).
    evalf_func(G3_evalf).
    eval_func(G3_eval).
    overloaded(2))
```

Referenced by GiNaC::G().

The documentation for this class was generated from the following files:

- inifcns.h
- inifcns_nstdsums.cpp

8.67 GiNaC::gcd_options Struct Reference

Flags to control the behavior of gcd() and friends.

```
#include <normal.h>
```

Public Types

- enum { no_heur_gcd = 2 , no_part_factored = 4 , use_sr_gcd = 8 }

8.67.1 Detailed Description

Flags to control the behavior of gcd() and friends.

8.67.2 Member Enumeration Documentation

8.67.2.1 anonymous enum

```
anonymous enum
```

Enumerator

no_heur_gcd	Usually GiNaC tries heuristic GCD first, because typically it's much faster than anything else. Even if heuristic algorithm fails, the overhead is negligible w.r.t. cost of computing the GCD by some other method. However, some people dislike it, so here's a flag which tells GiNaC to NOT use the heuristic algorithm.

no_part_factored	GiNaC tries to avoid expanding expressions when computing GCDs. This is a good idea, but some people dislike it. Hence the flag to disable special handling of partially factored polynomials. DON'T SET THIS unless you *really* know what are you doing!
use_sr_gcd	By default GiNaC uses modular GCD algorithm. Typically it's much faster than PRS (pseudo remainder sequence) algorithm. This flag forces GiNaC to use PRS algorithm

The documentation for this struct was generated from the following file:

- normal.h

8.68 GiNaC::gcdheu_failed Class Reference

Exception thrown by heur_gcd() to signal failure.

8.68.1 Detailed Description

Exception thrown by heur_gcd() to signal failure.

The documentation for this class was generated from the following file:

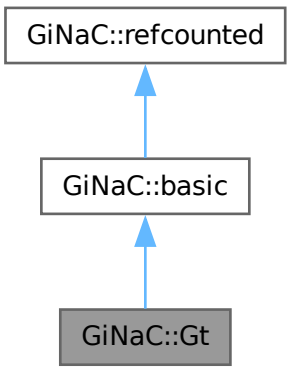
- normal.cpp

8.69 GiNaC::Gt Class Reference

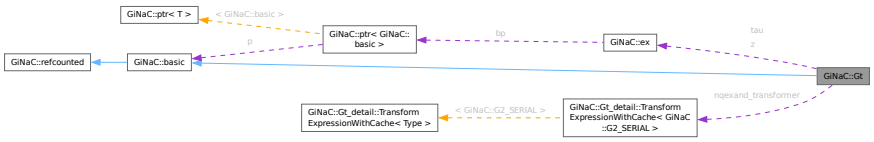
Elliptic Multiple Polylogarithm.

`#include <Gt.h>`

Inheritance diagram for GiNaC::Gt:



Collaboration diagram for GiNaC::Gt:



Classes

- struct kernel

Public Member Functions

- Gt (const ex &args, const ex &z, const ex &tau)
- const ex & getTau () const
- const ex & getZ () const
- const std::vector< kernel > & getArgs () const
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex evalf () const override

  Evaluate object numerically.
- size_t nops () const override

  Number of operands/members.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- bool has (const ex &other, unsigned options=0) const override

  Test for occurrence of a pattern.
- bool match (const ex &pattern, exmap &repl_lst) const override

  Check whether the expression matches a given pattern.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- ex zisToFundamental (const ex *points=nullptr) const
- ex regularise (const ex *points=nullptr) const
- matrix findMoebiusTransform (const ex *points=nullptr) const
- ex tauToFundamental (const ex *points=nullptr) const
- std::vector< ex > decomposeIntegrationPath (const ex *points=nullptr) const
- ex applyIntegrationPath (const std::vector< ex > &endpoints) const
- numeric qExpand (const ex *points=nullptr) const
- numeric evaluate (const ex *points=nullptr) const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

Default implementation of ex::to_rational().

- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Static Public Member Functions

- static ex ex_zisToFundamental (const ex &expr, const ex *points=nullptr)
- static ex ex_regularise (const ex &expr, const ex *points=nullptr)
- static ex ex_tauToFundamental (const ex &expr, const ex *points=nullptr)
- static ex ex_cutIntegrationPath (const ex &expr, const ex *points=nullptr)
- static numeric ex_qExpand (const ex &expr, const ex *points=nullptr)
- static ex ex_prepare (const ex &expr, const ex *points=nullptr)
- static numeric ex_evaluate (const ex &expr, const ex *points=nullptr)
- static lst lst_evaluate (const lst &list, const ex *points=nullptr)
- static ex ex_zisToFundamental (const ex &expr, const ex &points)
- static ex ex_regularise (const ex &expr, const ex &points)
- static ex ex_tauToFundamental (const ex &expr, const ex &points)
- static ex ex_cutIntegrationPath (const ex &expr, const ex &points)
- static numeric ex_qExpand (const ex &expr, const ex &points)
- static ex ex_prepare (const ex &expr, const ex &points)
- static numeric ex_evaluate (const ex &expr, const ex &points)
- static lst lst_evaluate (const lst &list, const ex &points)
- static ex apply_function_recursive (const ex &input, const std::function< ex(const Gt &)> &func)
- static void clear_polylog_cache ()

Static Public Attributes

- static double cutIntegrationPath_threshold_real = 0.4
- static double cutIntegrationPath_threshold_imag = 0.4
- static size_t qExpand_minOrder = 4
- static size_t qExpand_stepSize = 4
- static bool auto_clear_polylog_cache = false
- static bool enable_tauToFundamental = true

Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Member Functions

- Gt (std::vector< kernel > args, ex z, ex tau)
- void setArgs (const ex &args)

Static Private Member Functions

- static numeric to_numeric (const ex &expr, const ex *points=nullptr)
- static long to_integer (const ex &expr, const bool allow_negative)

Private Attributes

- size_t nargs
- std::vector< kernel > args
- ex z
- ex tau

Static Private Attributes

- static Gt_detail::TransformExpressionWithCache< G2_SERIAL > nqexand_transformer

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags
  of type status_flags
- unsigned hashvalue
  hash value

8.69.1 Detailed Description

Elliptic Multiple Polylogarithm.

8.69.2 Constructor & Destructor Documentation

8.69.2.1 Gt() [1/2]

```
GiNaC::Gt::Gt (
            const ex & args,
            const ex & z,
            const ex & tau)
```

References GiNaC::ex::evalf(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::is_a(), nargs, setArgs(), tau, and z.

Referenced by apply_function_recursive(), applyIntegrationPath(), eval(), ex_cutIntegrationPath(), ex_qExpand(), ex_regularise(), ex_tauToFundamental(), ex_zisToFundamental(), regularise(), subs(), tauToFundamental(), and zisToFundamental().

8.69.2.2 Gt() [2/2]

```
GiNaC::Gt::Gt (
            std::vector< kernel > args,
            ex z,
            ex tau)  [private]
```

References args, GiNaC::ex::evalf(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::is_a(), nargs, tau, and z.

8.69.3 Member Function Documentation

8.69.3.1 setArgs()

```
void GiNaC::Gt::setArgs (
            const ex & args)  [private]
```

References args, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::is_a(), nargs, GiNaC::ex::nops(), and GiNaC::ex::op().

Referenced by Gt(), and read_archive().

8.69.3.2 getTau()

```
const ex & GiNaC::Gt::getTau () const  [inline]
```

References GiNaC::basic::ex, and tau.

8.69.3.3 getZ()

```
const ex & GiNaC::Gt::getZ () const  [inline]
```

References GiNaC::basic::ex, and z.

8.69.3.4 getArgs()

```
const std::vector< kernel > & GiNaC::Gt::getArgs () const  [inline]
```

References args.

8.69.3.5 precedence()

```
unsigned GiNaC::Gt::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

8.69.3.6 eval()

```
ex GiNaC::Gt::eval () const  [override], [virtual]
```

Perform automatic non-interruptive term rewriting rules.

Reimplemented from GiNaC::basic.

References args, GiNaC::status_flags::evaluated, GiNaC::basic::ex, GiNaC::factorial(), GiNaC::basic::flags, Gt(), GiNaC::basic::hold(), nargs, GiNaC::pow(), tau, and z.

8.69.3.7 evalf()

```
ex GiNaC::Gt::evalf () const  [override], [virtual]
```

Evaluate object numerically.

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), args, GiNaC::dynallocate(), GiNaC::ex::evalf(), evaluate(), GiNaC::basic::ex, GiNaC::is_a(), nargs, tau, and z.

Referenced by qExpand().

8.69.3.8 nops()

```
size_t GiNaC::Gt::nops () const  [inline], [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.69.3.9 subs()

```
ex GiNaC::Gt::subs (
            const exmap & m,
            unsigned options = 0) const  [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References args, GiNaC::basic::ex, Gt(), nargs, GiNaC::ex::subs(), tau, and z.

8.69.3.10 calchash()

```
unsigned GiNaC::Gt::calchash () const  [override], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References args, GiNaC::status_flags::evaluated, GiNaC::basic::flags, GiNaC::basic::gethash(), GiNaC::ex::gethash(), GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, GiNaC::rotate_left(), GiNaC::basic::setflag(), tau, and z.

8.69.3.11 has()

```
bool GiNaC::Gt::has (
            const ex & pattern,
            unsigned options = 0) const  [override], [virtual]
```

Test for occurrence of a pattern.

An object 'has' a pattern if it matches the pattern itself or one of the children 'has' it. As a consequence (according to the definition of children) given e=x+y+z, e.has(x) is true but e.has(x+y) is false.

Reimplemented from GiNaC::basic.

References args, GiNaC::basic::ex, GiNaC::ex::has(), tau, and z.

8.69.3.12 match()

```
bool GiNaC::Gt::match (
            const ex & pattern,
            exmap & repl_lst) const  [override], [virtual]
```

Check whether the expression matches a given pattern.

For every wildcard object in the pattern, a pair with the wildcard as a key and matching expression as a value is added to repl_lst.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, and GiNaC::ex_to().

8.69.3.13 archive()

```
void GiNaC::Gt::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_ex(), GiNaC::container< C >::append(), nargs, tau, and z.

8.69.3.14 read_archive()

```
void GiNaC::Gt::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References args, GiNaC::basic::ex, GiNaC::archive_node::find_ex(), setArgs(), tau, and z.

8.69.3.15 zisToFundamental()

```
ex GiNaC::Gt::zisToFundamental (
            const ex * points = nullptr) const
```

References GiNaC::add::add(), args, GiNaC::class_info< OPT >::first, DEBUG_GT, GiNaC::basic::ex, GiNaC::factor(), GiNaC::factorial(), Gt(), GiNaC::I, GiNaC::numeric::imag(), GiNaC::is_zero(), nargs, GiNaC::ex::normal(), GiNaC::Pi, GiNaC::pow(), GiNaC::numeric::real(), tau, to_numeric(), and z.

Referenced by ex_zisToFundamental().

8.69.3.16 regularise()

```
ex GiNaC::Gt::regularise (
            const ex * points = nullptr) const
```

References GiNaC::add::add(), args, DEBUG_GT, GiNaC::basic::ex, GiNaC::factorial(), GiNaC::basic_multi_iterator< T >::get_vecto Gt(), GiNaC::multi_iterator_shuffle< T >::init(), GiNaC::is_zero(), nargs, GiNaC::basic_multi_iterator< T >::overflow(), GiNaC::pow(), tau, to_numeric(), and z.

Referenced by ex_regularise().

8.69.3.17 findMoebiusTransform()

```
matrix GiNaC::Gt::findMoebiusTransform (
            const ex * points = nullptr) const
```

References GiNaC::abs(), GiNaC::basic::ex, GiNaC::numeric::imag(), GiNaC::numeric::is_positive(), GiNaC::matrix::mul(), tau, GiNaC::numeric::to_cl_N(), and to_numeric().

Referenced by tauToFundamental().

8.69.3.18 tauToFundamental()

```
ex GiNaC::Gt::tauToFundamental (
            const ex * points = nullptr) const
```

References GiNaC::add::add(), args, GiNaC::binomial(), DEBUG_GT, evaluate(), GiNaC::basic::ex, GiNaC::factorial(), findMoebiusTransform(), Gt(), GiNaC::I, GiNaC::ex::is_zero(), GiNaC::is_zero(), GiNaC::log(), nargs, GiNaC::Gt::kernel::ni, GiNaC::basic::normal(), GiNaC::Pi, GiNaC::pow(), tau, to_numeric(), and z.

Referenced by ex_tauToFundamental().

8.69.3.19 decomposeIntegrationPath()

```
std::vector< ex > GiNaC::Gt::decomposeIntegrationPath (
            const ex * points = nullptr) const
```

References args, cutIntegrationPath_threshold_imag, cutIntegrationPath_threshold_real, DEBUG_GT, GiNaC::basic::diff(), GiNaC::basic::ex, GiNaC::numeric::imag(), GiNaC::numeric::is_integer(), GiNaC::numeric::real(), tau, GiNaC::numeric::to_double(), to_numeric(), and z.

Referenced by ex_cutIntegrationPath().

8.69.3.20 applyIntegrationPath()

```
ex GiNaC::Gt::applyIntegrationPath (
            const std::vector< ex > & endpoints) const
```

References args, GiNaC::Gt_detail::deconcatenate_path(), GiNaC::basic::ex, Gt(), and tau.

Referenced by ex_cutIntegrationPath().

8.69.3.21 qExpand()

```
numeric GiNaC::Gt::qExpand (
            const ex * points = nullptr) const
```

References GiNaC::abs(), GiNaC::add::add(), args, GiNaC::binomial(), GiNaC::class_info< OPT >::first, DEBUG_GT, GiNaC::Digits, evalf(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::exp(), GiNaC::factor(), GiNaC::factorial(), GiNaC::I, GiNaC::numeric::imag(), GiNaC::numeric::is_negative(), GiNaC::numeric::is_positive(), GiNaC::log(), nargs, nqexand_transformer, GiNaC::Pi, GiNaC::pow(), qExpand_minOrder, qExpand_stepSize, GiNaC::numeric::real(), GiNaC::real(), tau, GiNaC::numeric::to_double(), to_numeric(), z, and GiNaC::zeta().

Referenced by ex_qExpand().

8.69.3.22 evaluate()

```
numeric GiNaC::Gt::evaluate (
            const ex * points = nullptr) const
```

References GiNaC::basic::ex, and ex_evaluate().

Referenced by evalf(), and tauToFundamental().

8.69.3.23 ex_zisToFundamental() [1/2]

```
ex GiNaC::Gt::ex_zisToFundamental (
            const ex & expr,
            const ex * points = nullptr)  [static]
```

References apply_function_recursive(), GiNaC::basic::ex, Gt(), and zisToFundamental().

Referenced by ex_prepare(), and ex_zisToFundamental().

8.69.3.24 ex_regularise() [1/2]

```
ex GiNaC::Gt::ex_regularise (
            const ex & expr,
            const ex * points = nullptr)  [static]
```

References apply_function_recursive(), GiNaC::basic::ex, Gt(), and regularise().

Referenced by ex_prepare(), and ex_regularise().

8.69.3.25 ex_tauToFundamental() [1/2]

```
ex GiNaC::Gt::ex_tauToFundamental (
            const ex & expr,
            const ex * points = nullptr)  [static]
```

References apply_function_recursive(), GiNaC::basic::ex, Gt(), and tauToFundamental().

Referenced by ex_prepare(), and ex_tauToFundamental().

8.69.3.26 ex_cutIntegrationPath() [1/2]

```
ex GiNaC::Gt::ex_cutIntegrationPath (
            const ex & expr,
            const ex * points = nullptr)  [static]
```

References apply_function_recursive(), applyIntegrationPath(), decomposeIntegrationPath(), GiNaC::basic::ex, and Gt().

Referenced by ex_cutIntegrationPath(), and ex_prepare().

8.69.3.27 ex_qExpand() [1/2]

```
numeric GiNaC::Gt::ex_qExpand (
            const ex & expr,
            const ex * points = nullptr)  [static]
```

References apply_function_recursive(), GiNaC::basic::ex, Gt(), qExpand(), and to_numeric().

Referenced by ex_evaluate(), ex_qExpand(), and lst_evaluate().

8.69.3.28 ex_prepare() [1/2]

```
ex GiNaC::Gt::ex_prepare (
            const ex & expr,
            const ex * points = nullptr)  [static]
```

References DEBUG_GT, enable_tauToFundamental, GiNaC::basic::ex, ex_cutIntegrationPath(), ex_regularise(), ex_tauToFundamental(), and ex_zisToFundamental().

Referenced by ex_evaluate(), ex_prepare(), and lst_evaluate().

8.69.3.29 ex_evaluate() [1/2]

```
numeric GiNaC::Gt::ex_evaluate (
            const ex & expr,
            const ex * points = nullptr)  [static]
```

References auto_clear_polylog_cache, clear_polylog_cache(), DEBUG_GT, GiNaC::basic::ex, ex_prepare(), ex_qExpand(), and to_numeric().

Referenced by evaluate(), and ex_evaluate().

8.69.3.30 lst_evaluate() [1/2]

```
lst GiNaC::Gt::lst_evaluate (
            const lst & list,
            const ex * points = nullptr)  [static]
```

References GiNaC::container< C >::append(), auto_clear_polylog_cache, clear_polylog_cache(), DEBUG_GT, GiNaC::basic::ex, ex_prepare(), ex_qExpand(), GiNaC::ex_to(), GiNaC::container< C >::nops(), GiNaC::container< C >::op(), and to_numeric().

Referenced by lst_evaluate().

8.69.3.31 ex_zisToFundamental() [2/2]

```
ex GiNaC::Gt::ex_zisToFundamental (
            const ex & expr,
            const ex & points)  [static]
```

References GiNaC::basic::ex, and ex_zisToFundamental().

8.69.3.32 ex_regularise() [2/2]

```
ex GiNaC::Gt::ex_regularise (
            const ex & expr,
            const ex & points)  [static]
```

References GiNaC::basic::ex, and ex_regularise().

8.69.3.33 ex_tauToFundamental() [2/2]

```
ex GiNaC::Gt::ex_tauToFundamental (
            const ex & expr,
            const ex & points)  [static]
```

References GiNaC::basic::ex, and ex_tauToFundamental().

8.69.3.34 ex_cutIntegrationPath() [2/2]

```
ex GiNaC::Gt::ex_cutIntegrationPath (
            const ex & expr,
            const ex & points)  [static]
```

References GiNaC::basic::ex, and ex_cutIntegrationPath().

8.69.3.35 ex_qExpand() [2/2]

```
numeric GiNaC::Gt::ex_qExpand (
            const ex & expr,
            const ex & points)  [static]
```

References GiNaC::basic::ex, and ex_qExpand().

8.69.3.36 ex_prepare() [2/2]

```
ex GiNaC::Gt::ex_prepare (
            const ex & expr,
            const ex & points)  [static]
```

References GiNaC::basic::ex, and ex_prepare().

8.69.3.37 ex_evaluate() [2/2]

```
numeric GiNaC::Gt::ex_evaluate (
            const ex & expr,
            const ex & points)  [static]
```

References GiNaC::basic::ex, and ex_evaluate().

8.69.3.38 lst_evaluate() [2/2]

```
lst GiNaC::Gt::lst_evaluate (
            const lst & list,
            const ex & points)  [static]
```

References GiNaC::basic::ex, and lst_evaluate().

8.69.3.39 apply_function_recursive()

```
ex GiNaC::Gt::apply_function_recursive (
            const ex & input,
            const std::function< ex(const Gt &)> & func)  [static]
```

References GiNaC::basic::ex, GiNaC::ex_to(), and Gt().

Referenced by ex_cutIntegrationPath(), ex_qExpand(), ex_regularise(), ex_tauToFundamental(), and ex_zisToFundamental().

8.69.3.40 clear_polylog_cache()

```
void GiNaC::Gt::clear_polylog_cache ()  [static]
```

References nqexand_transformer.

Referenced by ex_evaluate(), and lst_evaluate().

8.69.3.41 do_print()

```
void GiNaC::Gt::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References args, nargs, GiNaC::print_context::s, tau, and z.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.69.3.42 do_print_latex()

```
void GiNaC::Gt::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

References args, nargs, GiNaC::print_context::s, tau, and z.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.69.3.43 to_numeric()

```
numeric GiNaC::Gt::to_numeric (
            const ex & expr,
            const ex * points = nullptr)  [static], [private]
```

References GiNaC::numeric::evalf(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::is_a(), and GiNaC::ex::subs().

Referenced by decomposeIntegrationPath(), ex_evaluate(), ex_qExpand(), findMoebiusTransform(), lst_evaluate(), qExpand(), regularise(), tauToFundamental(), and zisToFundamental().

8.69.3.44 to_integer()

```
long GiNaC::Gt::to_integer (
            const ex & expr,
            const bool allow_negative)  [static], [private]
```

References GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::is_a(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_nonneg_integer(), and GiNaC::numeric::to_long().

Referenced by GiNaC::Gt::kernel::kernel().

8.69.4 Member Data Documentation

8.69.4.1 cutIntegrationPath_threshold_real

```
double GiNaC::Gt::cutIntegrationPath_threshold_real = 0.4  [static]
```

Referenced by decomposeIntegrationPath().

8.69.4.2 cutIntegrationPath_threshold_imag

```
double GiNaC::Gt::cutIntegrationPath_threshold_imag = 0.4  [static]
```

Referenced by decomposeIntegrationPath().

8.69.4.3 qExpand_minOrder

```
size_t GiNaC::Gt::qExpand_minOrder = 4  [static]
```

Referenced by qExpand().

8.69.4.4 qExpand_stepSize

```
size_t GiNaC::Gt::qExpand_stepSize = 4  [static]
```

Referenced by qExpand().

8.69.4.5 auto_clear_polylog_cache

```
bool GiNaC::Gt::auto_clear_polylog_cache = false  [static]
```

Referenced by ex_evaluate(), and lst_evaluate().

8.69.4.6 enable_tauToFundamental

```
bool GiNaC::Gt::enable_tauToFundamental = true  [static]
```

Referenced by ex_prepare().

8.69.4.7 nargs

```
size_t GiNaC::Gt::nargs  [private]
```

Referenced by archive(), do_print(), do_print_latex(), eval(), evalf(), Gt(), Gt(), qExpand(), regularise(), setArgs(), subs(), tauToFundamental(), and zisToFundamental().

8.69.4.8 args

```
std::vector<kernel> GiNaC::Gt::args  [private]
```

Referenced by applyIntegrationPath(), calchash(), decomposeIntegrationPath(), do_print(), do_print_latex(), eval(), evalf(), getArgs(), Gt(), has(), qExpand(), read_archive(), regularise(), setArgs(), subs(), tauToFundamental(), and zisToFundamental().

8.69.4.9 z

```
ex GiNaC::Gt::z  [private]
```

Referenced by archive(), calchash(), decomposeIntegrationPath(), do_print(), do_print_latex(), eval(), evalf(), getZ(), Gt(), Gt(), has(), qExpand(), read_archive(), regularise(), subs(), tauToFundamental(), and zisToFundamental().

8.69.4.10 tau

```
ex GiNaC::Gt::tau  [private]
```

Referenced by applyIntegrationPath(), archive(), calchash(), decomposeIntegrationPath(), do_print(), do_print_latex(), eval(), evalf(), findMoebiusTransform(), getTau(), Gt(), Gt(), has(), qExpand(), read_archive(), regularise(), subs(), tauToFundamental(), and zisToFundamental().

8.69.4.11 nqexand_transformer

```
Gt_detail::TransformExpressionWithCache< G2_SERIAL > GiNaC::Gt::nqexand_transformer  [static],
[private]
```

Initial value:

```
                                                                = []()
{

    Digits.add_callback([](long change) -> void {
        if (change>0)Gt::clear_polylog_cache();
    });
    return Gt_detail::TransformExpressionWithCache<G2_SERIAL>{
        [](const ex& obj) {return obj.evalf();},
        [](const ex& obj) {return obj.evalf();}
    };
}()
```

Referenced by clear_polylog_cache(), and qExpand().

The documentation for this class was generated from the following files:

- Gt.h
- Gt.cpp

8.70 GiNaC::has_distance< T > Class Template Reference

SFINAE test for distance.

```
#include <utils_multi_iterator.h>
```

Public Types

- enum { value = sizeof(test<T>(0)) == sizeof(yes_type) }

Private Types

- typedef char yes_type[1]
- typedef char no_type[2]

Static Private Member Functions

- template<typename C>
  static yes_type & test (decltype(std::distance< C >))
- template<typename C>
  static no_type & test (...)

8.70.1 Detailed Description

template<typename T>
class GiNaC::has_distance< T >

SFINAE test for distance.

8.70.2 Member Typedef Documentation

8.70.2.1 yes_type

```
template<typename T>
typedef char GiNaC::has_distance< T >::yes_type[1]  [private]
```

8.70.2.2 no_type

```
template<typename T>
typedef char GiNaC::has_distance< T >::no_type[2]  [private]
```

8.70.3 Member Enumeration Documentation

8.70.3.1 anonymous enum

```
template<typename T>
anonymous enum
```

Enumerator

value	

8.70.4 Member Function Documentation

8.70.4.1 test() [1/2]

```
template<typename T>
template<typename C>
yes_type & GiNaC::has_distance< T >::test (
            decltype(std::distance< C >) )  [static], [private]
```

8.70.4.2 test() [2/2]

```
template<typename T>
template<typename C>
no_type & GiNaC::has_distance< T >::test (
              ...)  [static], [private]
```

The documentation for this class was generated from the following file:

- utils_multi_iterator.h

8.71 GiNaC::has_options Class Reference

Flags to control the behavior of has().

```
#include <flags.h>
```

Public Types

- enum { algebraic = 0x0001 }

8.71.1 Detailed Description

Flags to control the behavior of has().

8.71.2 Member Enumeration Documentation

8.71.2.1 anonymous enum

```
anonymous enum
```

Enumerator

algebraic	enable algebraic matching

The documentation for this class was generated from the following file:

- flags.h

8.72 std::hash< GiNaC::ex > Struct Reference

Specialization of std::hash() for ex objects.

```
#include <ex.h>
```

Public Member Functions

- std::size_t operator() (const GiNaC::ex &e) const noexcept

8.72.1 Detailed Description

Specialization of std::hash() for ex objects.

8.72.2 Member Function Documentation

8.72.2.1 operator()()

```
std::size_t std::hash< GiNaC::ex >::operator() (
            const GiNaC::ex & e) const  [inline], [noexcept]
```

The documentation for this struct was generated from the following file:

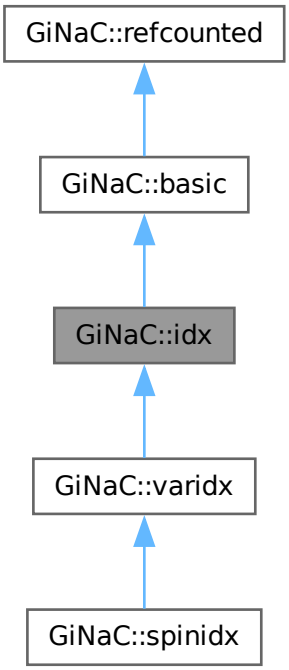
- ex.h

8.73 GiNaC::idx Class Reference

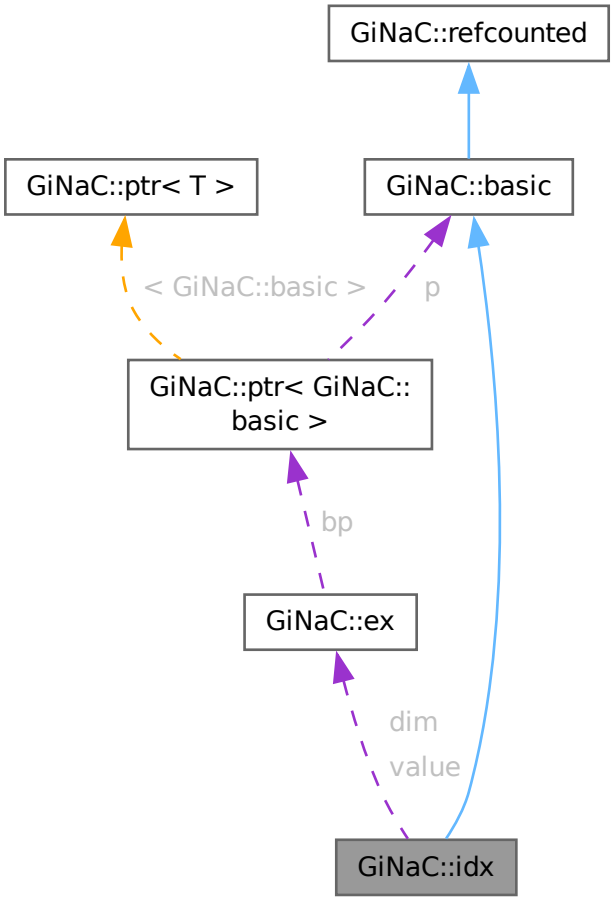
This class holds one index of an indexed object.

`#include <idx.h>`

Inheritance diagram for GiNaC::idx:



Collaboration diagram for GiNaC::idx:



Public Member Functions

- idx (const ex &v, const ex &dim)

  Construct index with given value and dimension.
- bool info (unsigned inf) const override

  Information about the object.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex map (map_function &f) const override

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- ex evalf () const override

  By default, basic::evalf would evaluate the index value but we don't want a.1 to become a.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.

- void archive (archive_node &n) const override

  Save (serialize) the object into archive node.
- void read_archive (const archive_node &n, lst &syms) override

  Load (deserialize) the object from an archive node.
- virtual bool is_dummy_pair_same_type (const basic &other) const

  Check whether the index forms a dummy index pair with another index of the same type.
- ex get_value () const

  Get value of index.
- bool is_numeric () const

  Check whether the index is numeric.
- bool is_symbolic () const

  Check whether the index is symbolic.
- ex get_dim () const

  Get dimension of index space.
- bool is_dim_numeric () const

  Check whether the dimension is numeric.
- bool is_dim_symbolic () const

  Check whether the dimension is symbolic.
- ex replace_dim (const ex &new_dim) const

  Make a new index with the same value but a different dimension.
- ex minimal_dim (const idx &other) const

  Return the minimum of the dimensions of this and another index.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual ex operator[] (const ex &index) const
- virtual ex operator[] (size_t i) const

- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for an index always returns 0.
- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void print_index (const print_context &c, unsigned level) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_csrc (const print_csrc &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex value

  Expression that constitutes the index (numeric or symbolic name).
- ex dim

  Dimension of space (can be symbolic or numeric).

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.73.1 Detailed Description

This class holds one index of an indexed object.

Indices can theoretically consist of any symbolic expression but they are usually only just a symbol (e.g. "mu", "i") or numeric (integer). Indices belong to a space with a certain numeric or symbolic dimension.

8.73.2 Constructor & Destructor Documentation

8.73.2.1 idx()

```
GiNaC::idx::idx (
            const ex & v,
            const ex & dim)  [explicit]
```

Construct index with given value and dimension.

Parameters

v	Value of index (numeric or symbolic)
dim	Dimension of index space (numeric or symbolic)

References dim, GiNaC::basic::ex, GiNaC::ex::info(), is_dim_numeric(), GiNaC::info_flags::posint, and value.

Referenced by GiNaC::expand_dummy_sum(), is_dummy_pair_same_type(), map(), match_same_type(), minimal_dim(), replace_dim(), and subs().

8.73.3 Member Function Documentation

8.73.3.1 info()

```
bool GiNaC::idx::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References GiNaC::info_flags::has_indices, and GiNaC::info_flags::idx.

8.73.3.2 nops()

```
size_t GiNaC::idx::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.73.3.3 op()

```
ex GiNaC::idx::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GINAC_ASSERT, and value.

8.73.3.4 map()

```
ex GiNaC::idx::map (
            map_function & f) const  [override], [virtual]
```

Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), GiNaC::basic::clearflag(), GiNaC::basic::duplicate(), GiNaC::basic::ex, GiNaC::status_flags::hash_calculated, idx(), and value.

8.73.3.5 evalf()

```
ex GiNaC::idx::evalf () const  [override], [virtual]
```

By default, basic::evalf would evaluate the index value but we don't want a.1 to become a.

(1.0).

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.73.3.6 subs()

```
ex GiNaC::idx::subs (
            const exmap & m,
            unsigned options = 0) const  [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), GiNaC::basic::clearflag(), GiNaC::basic::duplicate(), GiNaC::basic::ex, GiNaC::ex::find(), GiNaC::status_flags::hash_calculated, idx(), GiNaC::is_a(), GiNaC::subs_options::really_subs_idx, GiNaC::ex::subs(), and value.

8.73.3.7 archive()

```
void GiNaC::idx::archive (
            archive_node & n) const  [override], [virtual]
```

Save (serialize) the object into archive node.

Archive the object.

Losely speaking, this method turns an expression into a byte stream (which can be saved and restored later on, or sent via network, etc.)

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::spinidx, and GiNaC::varidx.

References GiNaC::archive_node::add_ex(), dim, and value.

8.73.3.8 read_archive()

```
void GiNaC::idx::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::spinidx, and GiNaC::varidx.

References dim, GiNaC::archive_node::find_ex(), and value.

8.73.3.9 derivative()

```
ex GiNaC::idx::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for an index always returns 0.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, and GiNaC::basic::ex.

8.73.3.10 match_same_type()

```
bool GiNaC::idx::match_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::spinidx, and GiNaC::varidx.

References GiNaC::basic::basic(), dim, GINAC_ASSERT, idx(), GiNaC::is_a(), and GiNaC::ex::is_equal().

8.73.3.11 calchash()

```
unsigned GiNaC::idx::calchash () const  [override], [protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References GiNaC::status_flags::evaluated, GiNaC::basic::flags, GiNaC::ex::gethash(), GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, GiNaC::rotate_left(), GiNaC::basic::setflag(), and value.

8.73.3.12 is_dummy_pair_same_type()

```
bool GiNaC::idx::is_dummy_pair_same_type (
            const basic & other) const  [virtual]
```

Check whether the index forms a dummy index pair with another index of the same type.

Reimplemented in GiNaC::spinidx, and GiNaC::varidx.

References GiNaC::basic::basic(), dim, idx(), GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), and value.

Referenced by GiNaC::is_dummy_pair().

8.73.3.13 get_value()

```
ex GiNaC::idx::get_value () const  [inline]
```

Get value of index.

References GiNaC::basic::ex, and value.

Referenced by GiNaC::matrix::eval_indexed(), GiNaC::minkmetric::eval_indexed(), GiNaC::spinmetric::eval_indexed(), and GiNaC::tensdelta::eval_indexed().

8.73.3.14 is_numeric()

```
bool GiNaC::idx::is_numeric () const  [inline]
```

Check whether the index is numeric.

References GiNaC::is_exactly_a(), and value.

8.73.3.15 is_symbolic()

```
bool GiNaC::idx::is_symbolic () const  [inline]
```

Check whether the index is symbolic.

References GiNaC::is_exactly_a(), and value.

Referenced by GiNaC::spinmetric::contract_with(), and GiNaC::tensor::replace_contr_index().

8.73.3.16 get_dim()

```
ex GiNaC::idx::get_dim () const  [inline]
```

Get dimension of index space.

References dim, and GiNaC::basic::ex.

Referenced by GiNaC::matrix::eval_indexed(), GiNaC::tensdelta::eval_indexed(), and GiNaC::tensmetric::eval_indexed().

8.73.3.17 is_dim_numeric()

```
bool GiNaC::idx::is_dim_numeric () const  [inline]
```

Check whether the dimension is numeric.

References dim, and GiNaC::is_exactly_a().

Referenced by idx().

8.73.3.18 is_dim_symbolic()

```
bool GiNaC::idx::is_dim_symbolic () const  [inline]
```

Check whether the dimension is symbolic.

References dim, and GiNaC::is_exactly_a().

8.73.3.19 replace_dim()

```
ex GiNaC::idx::replace_dim (
            const ex & new_dim) const
```

Make a new index with the same value but a different dimension.

References GiNaC::basic::clearflag(), dim, GiNaC::basic::duplicate(), GiNaC::basic::ex, GiNaC::status_flags::hash_calculated, and idx().

Referenced by GiNaC::tensdelta::eval_indexed(), GiNaC::tensmetric::eval_indexed(), and GiNaC::tensor::replace_contr_index().

8.73.3.20 minimal_dim()

```
ex GiNaC::idx::minimal_dim (
            const idx & other) const
```

Return the minimum of the dimensions of this and another index.

If this is undecidable, throw an exception.

References dim, GiNaC::basic::ex, idx(), and GiNaC::minimal_dim().

Referenced by GiNaC::tensdelta::eval_indexed(), GiNaC::tensmetric::eval_indexed(), and GiNaC::tensor::replace_contr_index().

8.73.3.21 print_index()

```
void GiNaC::idx::print_index (
            const print_context & c,
            unsigned level) const  [protected]
```

References dim, GiNaC::is_a(), GiNaC::is_exactly_a(), GiNaC::print_context::options, GiNaC::ex::print(), GiNaC::print_options::print_index_dimensions, GiNaC::print_context::s, and value.

Referenced by do_print(), GiNaC::spinidx::do_print(), GiNaC::varidx::do_print(), do_print_latex(), and GiNaC::spinidx::do_print_latex().

8.73.3.22 do_print()

```
void GiNaC::idx::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References print_index(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.73.3.23 do_print_csrc()

```
void GiNaC::idx::do_print_csrc (
            const print_csrc & c,
            unsigned level) const  [protected]
```

References GiNaC::ex_to(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::ex::print(), GiNaC::print_context::s, and value.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.73.3.24 do_print_latex()

```
void GiNaC::idx::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

References print_index(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.73.3.25 do_print_tree()

```
void GiNaC::idx::do_print_tree (
            const print_tree & c,
            unsigned level) const  [protected]
```

References GiNaC::print_tree::delta_indent, dim, GiNaC::basic::flags, GiNaC::basic::hashvalue, GiNaC::ex::print(), GiNaC::print_context::s, and value.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.73.4 Member Data Documentation

8.73.4.1 value

```
ex GiNaC::idx::value  [protected]
```

Expression that constitutes the index (numeric or symbolic name).

Referenced by archive(), calchash(), do_print_csrc(), do_print_tree(), GiNaC::spinidx::do_print_tree(), GiNaC::varidx::do_print_tree(), get_value(), idx(), is_dummy_pair_same_type(), is_numeric(), is_symbolic(), map(), op(), print_index(), read_archive(), and subs().

8.73.4.2 dim

```
ex GiNaC::idx::dim  [protected]
```

Dimension of space (can be symbolic or numeric).

Referenced by archive(), do_print_tree(), GiNaC::spinidx::do_print_tree(), GiNaC::varidx::do_print_tree(), get_dim(), idx(), is_dim_numeric(), is_dim_symbolic(), is_dummy_pair_same_type(), match_same_type(), minimal_dim(), print_index(), read_archive(), and replace_dim().

The documentation for this class was generated from the following files:

- idx.h
- idx.cpp

8.74 GiNaC::idx_is_equal_ignore_dim Struct Reference

Public Member Functions

- bool operator() (const ex &lh, const ex &rh) const

8.74.1 Member Function Documentation

8.74.1.1 operator()()

```
bool GiNaC::idx_is_equal_ignore_dim::operator() (
            const ex & lh,
            const ex & rh) const  [inline]
```

References GiNaC::ex_to(), and GiNaC::ex::is_equal().

The documentation for this struct was generated from the following file:

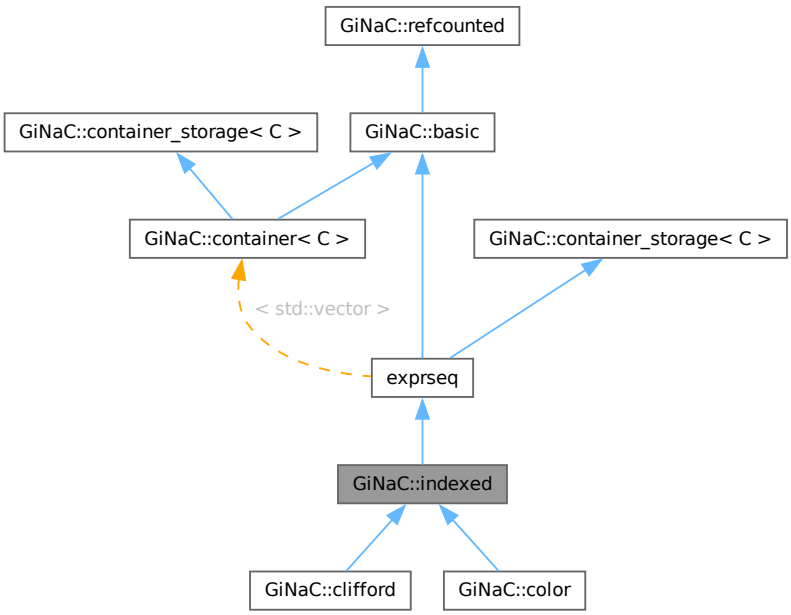
- indexed.cpp

8.75 GiNaC::indexed Class Reference

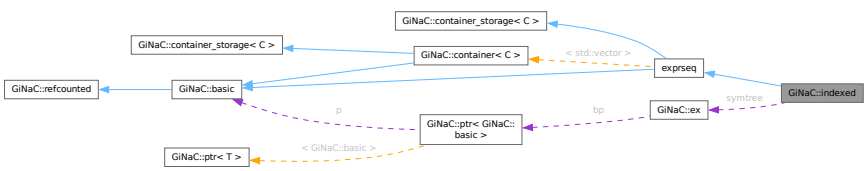
This class holds an indexed expression.

`#include <indexed.h>`

Inheritance diagram for GiNaC::indexed:



Collaboration diagram for GiNaC::indexed:



Public Member Functions

- indexed (const ex &b)

  Construct indexed object with no index.
- indexed (const ex &b, const ex &i1)

  Construct indexed object with one index.
- indexed (const ex &b, const ex &i1, const ex &i2)

  Construct indexed object with two indices.
- indexed (const ex &b, const ex &i1, const ex &i2, const ex &i3)

  Construct indexed object with three indices.
- indexed (const ex &b, const ex &i1, const ex &i2, const ex &i3, const ex &i4)

  Construct indexed object with four indices.
- indexed (const ex &b, const symmetry &symm, const ex &i1, const ex &i2)

  Construct indexed object with two indices and a specified symmetry.
- indexed (const ex &b, const symmetry &symm, const ex &i1, const ex &i2, const ex &i3)

  Construct indexed object with three indices and a specified symmetry.
- indexed (const ex &b, const symmetry &symm, const ex &i1, const ex &i2, const ex &i3, const ex &i4)

  Construct indexed object with four indices and a specified symmetry.
- indexed (const ex &b, const exvector &iv)

  Construct indexed object with a specified vector of indices.
- indexed (const ex &b, const symmetry &symm, const exvector &iv)

  Construct indexed object with a specified vector of indices and symmetry.
- indexed (const symmetry &symm, const exprseq &es)
- indexed (const symmetry &symm, const exvector &v)
- indexed (const symmetry &symm, exvector &&v)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex real_part () const override
- ex imag_part () const override
- exvector get_free_indices () const override

  Return a vector containing the free indices of an expression.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- bool all_index_values_are (unsigned inf) const

  Check whether all index values have a certain property.
- exvector get_indices () const

  Return a vector containing the object's indices.
- exvector get_dummy_indices () const

  Return a vector containing the dummy indices of the object, if any.
- exvector get_dummy_indices (const indexed &other) const

  Return a vector containing the dummy indices in the contraction with another indexed object.
- bool has_dummy_index_for (const ex &i) const

  Check whether the object has an index that forms a dummy index pair with a given index.
- ex get_symmetry () const

  Return symmetry properties.

Public Member Functions inherited from GiNaC::container< std::vector >

- container (STLT const &s)
- container (STLT &&v)
- container (exvector::const_iterator b, exvector::const_iterator e)
- container (std::initializer_list< ex > il)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- container & prepend (const ex &b)

  Add element at front.
- container & append (const ex &b)

  Add element at back.
- container & remove_first ()

  Remove first element.
- container & remove_last ()

  Remove last element.
- container & remove_all ()

  Remove all elements.
- container & sort ()

  Sort elements.
- container & unique ()

  Remove adjacent duplicate elements.
- const_iterator begin () const
- const_iterator end () const
- const_reverse_iterator rbegin () const
- const_reverse_iterator rend () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

Output to stream.

- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().

- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for an indexed object always returns 0.
- ex thiscontainer (const exvector &v) const override

  Similar to duplicate(), but with a preset sequence.
- ex thiscontainer (exvector &&v) const override

  Similar to duplicate(), but with a preset sequence (which gets pilfered).
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- void printindices (const print_context &c, unsigned level) const
- void print_indexed (const print_context &c, const char *openbrace, const char *closebrace, unsigned level) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void validate () const

  Check whether all indices are of class idx and validate the symmetry tree.

Protected Member Functions inherited from GiNaC::container< std::vector >

- ex conjugate () const override
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- virtual void printseq (const print_context &c, char openbracket, char delim, char closebracket, unsigned this↩_precedence, unsigned upper_precedence=0) const

  Print sequence of contained elements.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python (const print_python &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const
- STLT subschildren (const exmap &m, unsigned options=0) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Member Functions inherited from GiNaC::container_storage< C >

- container_storage ()
- container_storage (size_t n, const ex &e)
- container_storage (std::initializer_list< ex > il)
- template<class In>
  container_storage (In b, In e)
- void reserve (size_t)
- ~container_storage ()
- void reserve (size_t n)
- void reserve (std::vector< ex > &v, size_t n)

Protected Attributes

- ex symtree

  Index symmetry (tree of symmetry objects).

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

Protected Attributes inherited from GiNaC::container_storage< C >

- STLT seq

Friends

- ex simplify_indexed (const ex &e, exvector &free_indices, exvector &dummy_indices, const scalar_products &sp)

  Simplify indexed expression, return list of free indices.
- ex simplify_indexed_product (const ex &e, exvector &free_indices, exvector &dummy_indices, const scalar_products &sp)

  Simplify product of indexed expressions (commutative, noncommutative and simple squares), return list of free indices.
- bool reposition_dummy_indices (ex &e, exvector &variant_dummy_indices, exvector &moved_indices)

  Raise/lower dummy indices in a single indexed objects to canonicalize their variance.

Additional Inherited Members

Public Types inherited from GiNaC::container< std::vector >

- typedef STLT::const_iterator const_iterator
- typedef STLT::const_reverse_iterator const_reverse_iterator

Protected Types inherited from GiNaC::container< std::vector >

- typedef container_storage< std::vector >::STLT STLT

Protected Types inherited from GiNaC::container_storage< C >

- typedef C< ex > STLT

Static Protected Member Functions inherited from GiNaC::container< std::vector >

- static unsigned get_default_flags ()
- static char get_open_delim ()
- static char get_close_delim ()

Static Protected Member Functions inherited from GiNaC::container_storage< C >

- static void reserve (STLT &, size_t)

8.75.1 Detailed Description

This class holds an indexed expression.

It consists of a 'base' expression (the expression being indexed) which can be accessed as op(0), and n (n >= 0) indices (all of class idx), accessible as op(1)..op(n).

8.75.2 Constructor & Destructor Documentation

8.75.2.1 indexed() [1/13]

```
GiNaC::indexed::indexed (
            const ex & b)
```

Construct indexed object with no index.

Parameters

b	Base expression

References indexed(), GiNaC::not_symmetric(), symtree, and validate().

Referenced by GiNaC::matrix::add_indexed(), GiNaC::base_and_index(), GiNaC::clifford_unit(), GiNaC::color_d(), GiNaC::color_f(), GiNaC::matrix::contract_with(), GiNaC::delta_tensor(), GiNaC::dirac_gamma(), GiNaC::dirac_slash(), GiNaC::epsilon_tensor(), GiNaC::epsilon_tensor(), eval(), GiNaC::clifford::eval_ncmul(), get_dummy_indices(), GiNaC::clifford::get_metric(), indexed(), GiNaC::lorentz_eps(), GiNaC::lorentz_g(), GiNaC::lst_to_clifford(), GiNaC::metric_tensor(), GiNaC::matrix::scalar_mul_indexed(), simplify_indexed, GiNaC::spinor_metric(), thiscontainer(), and thiscontainer().

8.75.2.2 indexed() [2/13]

```
GiNaC::indexed::indexed (
            const ex & b,
            const ex & i1)
```

Construct indexed object with one index.

The index must be of class idx.

Parameters

b	Base expression
i1	The index

References GiNaC::not_symmetric(), symtree, and validate().

8.75.2.3 indexed() [3/13]

```
GiNaC::indexed::indexed (
            const ex & b,
            const ex & i1,
            const ex & i2)
```

Construct indexed object with two indices.

The indices must be of class idx.

Parameters

b	Base expression
i1	First index
i2	Second index

References GiNaC::not_symmetric(), symtree, and validate().

8.75.2.4 indexed() [4/13]

```
GiNaC::indexed::indexed (
              const ex & b,
              const ex & i1,
              const ex & i2,
              const ex & i3)
```

Construct indexed object with three indices.

The indices must be of class idx.

Parameters

b	Base expression
i1	First index
i2	Second index
i3	Third index

References GiNaC::not_symmetric(), symtree, and validate().

8.75.2.5 indexed() [5/13]

```
GiNaC::indexed::indexed (
              const ex & b,
              const ex & i1,
              const ex & i2,
              const ex & i3,
              const ex & i4)
```

Construct indexed object with four indices.

The indices must be of class idx.

Parameters

b	Base expression
i1	First index
i2	Second index
i3	Third index

i4	Fourth index

References GiNaC::not_symmetric(), symtree, and validate().

8.75.2.6 indexed() [6/13]

```
GiNaC::indexed::indexed (
              const ex & b,
              const symmetry & symm,
              const ex & i1,
              const ex & i2)
```

Construct indexed object with two indices and a specified symmetry.

The indices must be of class idx.

Parameters

b	Base expression
symm	Symmetry of indices
i1	First index
i2	Second index

References symtree, and validate().

8.75.2.7 indexed() [7/13]

```
GiNaC::indexed::indexed (
              const ex & b,
              const symmetry & symm,
              const ex & i1,
              const ex & i2,
              const ex & i3)
```

Construct indexed object with three indices and a specified symmetry.

The indices must be of class idx.

Parameters

b	Base expression
symm	Symmetry of indices
i1	First index
i2	Second index
i3	Third index

References symtree, and validate().

8.75.2.8 indexed() [8/13]

```
GiNaC::indexed::indexed (
              const ex & b,
              const symmetry & symm,
              const ex & i1,
              const ex & i2,
              const ex & i3,
              const ex & i4)
```

Construct indexed object with four indices and a specified symmetry.

The indices must be of class idx.

Parameters

b	Base expression
symm	Symmetry of indices
i1	First index
i2	Second index
i3	Third index
i4	Fourth index

References symtree, and validate().

8.75.2.9 indexed() [9/13]

```
GiNaC::indexed::indexed (
              const ex & b,
              const exvector & iv)
```

Construct indexed object with a specified vector of indices.

The indices must be of class idx.

Parameters

b	Base expression
iv	Vector of indices

References GiNaC::not_symmetric(), GiNaC::container_storage< C >::seq, symtree, and validate().

8.75.2.10 indexed() [10/13]

```
GiNaC::indexed::indexed (
            const ex & b,
            const symmetry & symm,
            const exvector & iv)
```

Construct indexed object with a specified vector of indices and symmetry.

The indices must be of class idx.

Parameters

b	Base expression
symm	Symmetry of indices
iv	Vector of indices

References GiNaC::container_storage< C >::seq, symtree, and validate().

8.75.2.11 indexed() [11/13]

```
GiNaC::indexed::indexed (
            const symmetry & symm,
            const exprseq & es)
```

References symtree.

8.75.2.12 indexed() [12/13]

```
GiNaC::indexed::indexed (
            const symmetry & symm,
            const exvector & v)
```

References symtree.

8.75.2.13 indexed() [13/13]

```
GiNaC::indexed::indexed (
            const symmetry & symm,
            exvector && v)
```

References symtree.

8.75.3 Member Function Documentation

8.75.3.1 precedence()

```
unsigned GiNaC::indexed::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::container< std::vector >.

Referenced by print_indexed().

8.75.3.2 info()

```
bool GiNaC::indexed::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::info_flags::has_indices, GiNaC::info_flags::indexed, and GiNaC::container_storage< C >::seq.

Referenced by imag_part(), and real_part().

8.75.3.3 eval()

```
ex GiNaC::indexed::eval () const  [override], [virtual]
```

Perform automatic non-interruptive term rewriting rules.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::canonicalize(), GiNaC::ex::ex(), GiNaC::ex_to(), GINAC_ASSERT, indexed(), GiNaC::is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::ex::nops(), GiNaC::ex::op(), GiNaC::container_storage< C >::seq, symtree, and thiscontainer().

8.75.3.4 real_part()

```
ex GiNaC::indexed::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::container< std::vector >.

References info(), GiNaC::container< std::vector >::op(), and GiNaC::info_flags::real.

8.75.3.5 imag_part()

```
ex GiNaC::indexed::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::container< std::vector >.

References info(), GiNaC::container< std::vector >::op(), and GiNaC::info_flags::real.

8.75.3.6 get_free_indices()

```
exvector GiNaC::indexed::get_free_indices () const  [override], [virtual]
```

Return a vector containing the free indices of an expression.

Reimplemented from GiNaC::basic.

References GiNaC::find_free_and_dummy(), and GiNaC::container_storage< C >::seq.

Referenced by get_dummy_indices().

8.75.3.7 archive()

```
void GiNaC::indexed::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) indexed object into archive.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::archive_node::add_ex(), and symtree.

8.75.3.8 read_archive()

```
void GiNaC::indexed::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) indexed object from archive.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::ex_to(), GiNaC::archive_node::find_ex(), GiNaC::archive_node::find_unsigned(), GiNaC::not_symmetric(), GiNaC::container_storage< C >::seq, GiNaC::sy_anti(), GiNaC::sy_symm(), symtree, and validate().

8.75.3.9 derivative()

```
ex GiNaC::indexed::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for an indexed object always returns 0.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::_ex0.

8.75.3.10 thiscontainer() [1/2]

```
ex GiNaC::indexed::thiscontainer (
            const exvector & v) const  [override], [protected], [virtual]
```

Similar to duplicate(), but with a preset sequence.

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::ex_to(), indexed(), and symtree.

Referenced by eval(), and expand().

8.75.3.11 thiscontainer() [2/2]

```
ex GiNaC::indexed::thiscontainer (
            exvector && v) const  [override], [protected], [virtual]
```

Similar to duplicate(), but with a preset sequence (which gets pilfered).

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::ex_to(), indexed(), and symtree.

8.75.3.12 return_type()

```
unsigned GiNaC::indexed::return_type () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative, GiNaC::is_a(), GiNaC::container< std::vector >::op(), and GiNaC::ex::return_type().

8.75.3.13 return_type_tinfo()

```
return_type_t GiNaC::indexed::return_type_tinfo () const  [inline], [override], [protected],
[virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::container< std::vector >::op(), and GiNaC::ex::return_type_tinfo().

8.75.3.14 expand()

```
ex GiNaC::indexed::expand (
            unsigned options = 0) const  [override], [protected], [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::are_ex_trivially_equal(), GiNaC::ex_to(), GiNaC::ex::expand(), GiNaC::expand_options::expand_inc
GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::ex::nops(), GiNaC::ex::op(), GiNaC::container_storage< C >::seq,
and thiscontainer().

8.75.3.15 all_index_values_are()

```
bool GiNaC::indexed::all_index_values_are (
            unsigned inf) const
```

Check whether all index values have a certain property.

See also

class info_flags

References GiNaC::container_storage< C >::seq.

8.75.3.16 get_indices()

```
exvector GiNaC::indexed::get_indices () const
```

Return a vector containing the object's indices.

References GINAC_ASSERT, and GiNaC::container_storage< C >::seq.

8.75.3.17 get_dummy_indices() [1/2]

```
exvector GiNaC::indexed::get_dummy_indices () const
```

Return a vector containing the dummy indices of the object, if any.

References GiNaC::find_free_and_dummy(), and GiNaC::container_storage< C >::seq.

8.75.3.18 get_dummy_indices() [2/2]

```
exvector GiNaC::indexed::get_dummy_indices (
            const indexed & other) const
```

Return a vector containing the dummy indices in the contraction with another indexed object.

This is symmetric: a.get_dummy_indices(b) == b.get_dummy_indices(a)

References GiNaC::find_dummy_indices(), get_free_indices(), and indexed().

8.75.3.19 has_dummy_index_for()

```
bool GiNaC::indexed::has_dummy_index_for (
            const ex & i) const
```

Check whether the object has an index that forms a dummy index pair with a given index.

References GiNaC::is_dummy_pair(), and GiNaC::container_storage< C >::seq.

8.75.3.20 get_symmetry()

```
ex GiNaC::indexed::get_symmetry () const  [inline]
```

Return symmetry properties.

References symtree.

Referenced by GiNaC::clifford::get_metric().

8.75.3.21 printindices()

```
void GiNaC::indexed::printindices (
            const print_context & c,
            unsigned level) const  [protected]
```

References GiNaC::class_info< OPT >::first, GiNaC::ex_to(), GiNaC::is_a(), GiNaC::print_context::s, and GiNaC::container_storage< C >::seq.

Referenced by GiNaC::clifford::do_print_dflt(), GiNaC::clifford::do_print_tree(), do_print_tree(), and print_indexed().

8.75.3.22 print_indexed()

```
void GiNaC::indexed::print_indexed (
            const print_context & c,
            const char * openbrace,
            const char * closebrace,
            unsigned level) const  [protected]
```

References precedence(), printindices(), GiNaC::print_context::s, and GiNaC::container_storage< C >::seq.

Referenced by do_print(), and do_print_latex().

8.75.3.23 do_print()

```
void GiNaC::indexed::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References print_indexed().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.75.3.24 do_print_latex()

```
void GiNaC::indexed::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

References print_indexed().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.75.3.25 do_print_tree()

```
void GiNaC::indexed::do_print_tree (
            const print_tree & c,
            unsigned level) const  [protected]
```

References GiNaC::print_tree::delta_indent, GiNaC::basic::flags, GiNaC::basic::hashvalue, printindices(), GiNaC::print_context::s, GiNaC::container_storage< C >::seq, and symtree.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.75.3.26 validate()

```
void GiNaC::indexed::validate () const  [protected]
```

Check whether all indices are of class idx and validate the symmetry tree.

This function is used internally to make sure that all constructed indexed objects really carry indices and not some other classes.

References GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_a(), GiNaC::is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::container_storage< C >::seq, symtree, and validate().

Referenced by indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), read_archive(), and validate().

8.75.4 Friends And Related Symbol Documentation

8.75.4.1 simplify_indexed

```
ex simplify_indexed (
            const ex & e,
            exvector & free_indices,
            exvector & dummy_indices,
            const scalar_products & sp)  [friend]
```

Simplify indexed expression, return list of free indices.

References GiNaC::_ex2, GiNaC::class_info< OPT >::first, GiNaC::basic::coeff(), GiNaC::ex::coeff(), GiNaC::dynallocate(), GiNaC::ex_to(), GiNaC::ex::expand(), GiNaC::find_free_and_dummy(), GiNaC::find_variant_indices(), GiNaC::hasindex(), GiNaC::idx_symmetrization(), indexed(), GiNaC::indices_consistent(), GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::ex::nops(), GiNaC::ex::op(), GiNaC::rename_dummy_indices(), reposition_dummy_indices, GiNaC::container_storage< C >::seq, simplify_indexed, simplify_indexed_product, GiNaC::symminfo::symminfo(), and GiNaC::terminfo::terminfo().

Referenced by GiNaC::clifford::get_metric(), GiNaC::clifford::same_metric(), simplify_indexed, and simplify_indexed_product.

8.75.4.2 simplify_indexed_product

```
ex simplify_indexed_product (
             const ex & e,
             exvector & free_indices,
             exvector & dummy_indices,
             const scalar_products & sp)  [friend]
```

Simplify product of indexed expressions (commutative, noncommutative and simple squares), return list of free indices.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::return_types::commutative, GiNaC::scalar_products::evaluate(), GiNaC::ex::ex(), GiNaC::ex_to(), GiNaC::find_free_and_dummy(), GiNaC::find_variant_indices(), GINAC_ASSERT, GiNaC::idx_symmetrization(), GiNaC::is_a(), GiNaC::scalar_products::is_defined(), GiNaC::is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::minimal_dim(), GiNaC::mul::mul(), GiNaC::ncmul::ncmul(), GiNaC::ex::nops(), GiNaC::ex::op(), GiNaC::product_to_exvector(), GiNaC::rename_dummy_indices(), reposition_dummy_indices, simplify_indexed, and simplify_indexed_product.

Referenced by simplify_indexed, and simplify_indexed_product.

8.75.4.3 reposition_dummy_indices

```
bool reposition_dummy_indices (
             ex & e,
             exvector & variant_dummy_indices,
             exvector & moved_indices)  [friend]
```

Raise/lower dummy indices in a single indexed objects to canonicalize their variance.

Parameters

e	Object to work on
variant_dummy_indices	The set of indices that might need repositioning (will be changed by this function)
moved_indices	The set of indices that have been repositioned (will be changed by this function)

Returns

true if 'e' was changed

References GiNaC::ex_to(), GiNaC::is_a(), GiNaC::is_dummy_pair(), GiNaC::is_exactly_a(), GiNaC::subs_options::no_pattern, GiNaC::ex::nops(), GiNaC::ex::op(), reposition_dummy_indices, GiNaC::container_storage< C >::seq, and GiNaC::ex::subs().

Referenced by reposition_dummy_indices, simplify_indexed, and simplify_indexed_product.

8.75.5 Member Data Documentation

8.75.5.1 symtree

```
ex GiNaC::indexed::symtree  [protected]
```

Index symmetry (tree of symmetry objects).

Referenced by archive(), GiNaC::clifford::do_print_tree(), do_print_tree(), eval(), get_symmetry(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), indexed(), read_archive(), thiscontainer(), thiscontainer(), and validate().

The documentation for this class was generated from the following files:

- indexed.h
- indexed.cpp

8.76 GiNaC::info_flags Class Reference

Possible attributes an object can have.

`#include <flags.h>`

Public Types

- enum {
  numeric , real , rational , integer ,
  crational , cinteger , positive , negative ,
  nonnegative , posint , negint , nonnegint ,
  even , odd , prime , relation ,
  relation_equal , relation_not_equal , relation_less , relation_less_or_equal ,
  relation_greater , relation_greater_or_equal , symbol , list ,
  exprseq , polynomial , integer_polynomial , cinteger_polynomial ,
  rational_polynomial , crational_polynomial , rational_function , indexed ,
  has_indices , idx , expanded , indefinite }

8.76.1 Detailed Description

Possible attributes an object can have.

8.76.2 Member Enumeration Documentation

8.76.2.1 anonymous enum

`anonymous enum`

Enumerator

numeric	
real	
rational	
integer	
crational	
cinteger	
positive	
negative	
nonnegative	
posint	
negint	
nonnegint	
even	
odd	

prime	
relation	
relation_equal	
relation_not_equal	
relation_less	
relation_less_or_equal	
relation_greater	
relation_greater_or_equal	
symbol	
list	
exprseq	
polynomial	
integer_polynomial	
cinteger_polynomial	
rational_polynomial	
crational_polynomial	
rational_function	
indexed	
has_indices	
idx	
expanded	
indefinite	

The documentation for this class was generated from the following file:

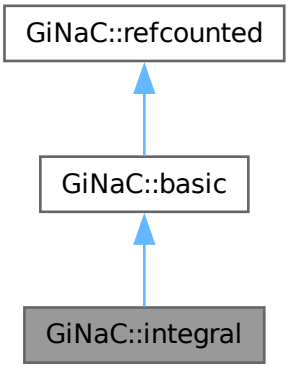
- flags.h

8.77 GiNaC::integral Class Reference

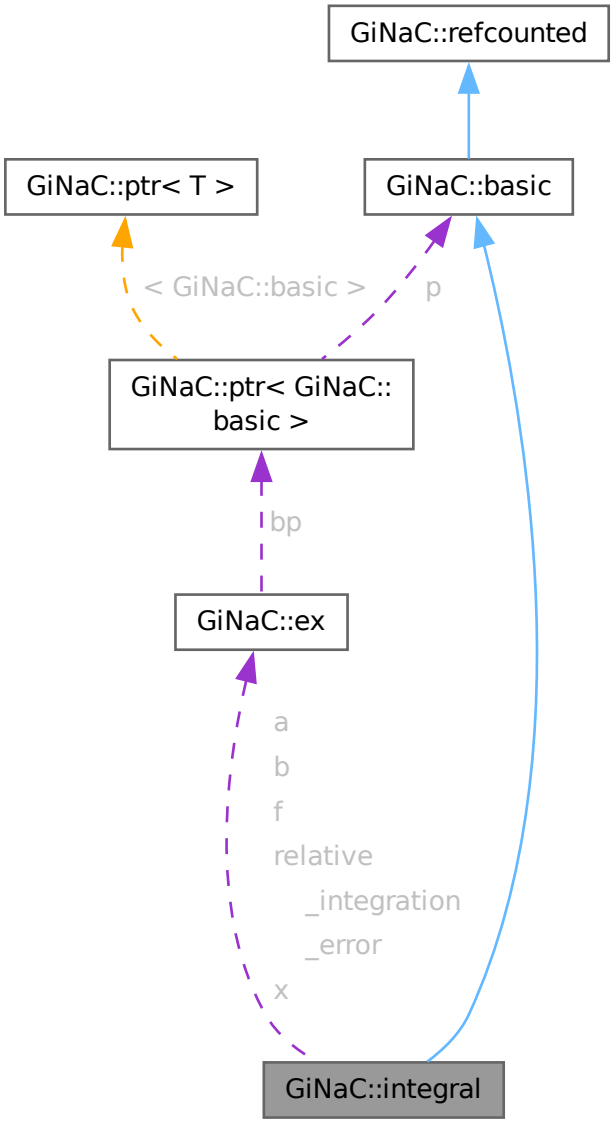
Symbolic integral.

`#include <integral.h>`

Inheritance diagram for GiNaC::integral:



Collaboration diagram for GiNaC::integral:



Public Member Functions

- integral (const ex &x_, const ex &a_, const ex &b_, const ex &f_)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex evalf () const override

  Evaluate object numerically.
- int degree (const ex &s) const override

Return degree of highest power in object s.

- int ldegree (const ex &s) const override

  Return degree of lowest power in object s.
- ex eval_ncmul (const exvector &v) const override
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- exvector get_free_indices () const override

  Return a vector containing the free indices of an expression.
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- ex conjugate () const override
- ex eval_integ () const override

  Evaluate integrals, if result is known.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Static Public Attributes

- static int max_integration_level = 15
- static ex relative_integration_error = 1e-8

Protected Member Functions

- ex derivative (const symbol &s) const override

  Default implementation of ex::diff().
- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Attributes

- ex x
- ex a
- ex b
- ex f

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags
  of type status_flags
- unsigned hashvalue
  hash value

8.77.1 Detailed Description

Symbolic integral.

8.77.2 Constructor & Destructor Documentation

8.77.2.1 integral()

```
GiNaC::integral::integral (
            const ex & x_,
            const ex & a_,
            const ex & b_,
            const ex & f_)
```

References a, b, GiNaC::basic::ex, f, integral(), GiNaC::is_a(), and x.

Referenced by GiNaC::adaptivesimpson(), derivative(), expand(), integral(), and series().

8.77.3 Member Function Documentation

8.77.3.1 precedence()

```
unsigned GiNaC::integral::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

Referenced by do_print_latex().

8.77.3.2 eval()

```
ex GiNaC::integral::eval () const  [override], [virtual]
```

Perform automatic non-interruptive term rewriting rules.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, a, b, GiNaC::status_flags::evaluated, GiNaC::basic::ex, f, GiNaC::basic::flags, GiNaC::ex::has(), GiNaC::haswild(), GiNaC::basic::hold(), and x.

8.77.3.3 evalf()

```
ex GiNaC::integral::evalf () const  [override], [virtual]
```

Evaluate object numerically.

Reimplemented from GiNaC::basic.

References a, GiNaC::adaptivesimpson(), GiNaC::are_ex_trivially_equal(), b, GiNaC::dynallocate(), GiNaC::ex::evalf(), GiNaC::basic::ex, f, GiNaC::is_exactly_a(), GiNaC::ex::subs(), and x.

8.77.3.4 degree()

```
int GiNaC::integral::degree (
            const ex & s) const  [override], [virtual]
```

Return degree of highest power in object s.

Reimplemented from GiNaC::basic.

References a, b, GiNaC::ex::degree(), GiNaC::basic::ex, and f.

8.77.3.5 ldegree()

```
int GiNaC::integral::ldegree (
            const ex & s) const  [override], [virtual]
```

Return degree of lowest power in object s.

Reimplemented from GiNaC::basic.

References a, b, GiNaC::basic::ex, and f.

8.77.3.6 eval_ncmul()

```
ex GiNaC::integral::eval_ncmul (
            const exvector & v) const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::ex::eval_ncmul(), GiNaC::basic::ex, and f.

8.77.3.7 nops()

```
size_t GiNaC::integral::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.77.3.8 op()

```
ex GiNaC::integral::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References a, b, GiNaC::basic::ex, f, GINAC_ASSERT, and x.

8.77.3.9 let_op()

```
ex & GiNaC::integral::let_op (
            size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References a, b, GiNaC::basic::ensure_if_modifiable(), GiNaC::basic::ex, f, and x.

8.77.3.10 expand()

```
ex GiNaC::integral::expand (
            unsigned options = 0) const  [override], [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented from GiNaC::basic.

References a, GiNaC::add::add(), GiNaC::are_ex_trivially_equal(), b, GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex::ex(), GiNaC::ex::expand(), GiNaC::status_flags::expanded, f, GiNaC::basic::flags, GiNaC::ex::has(), integral(), GiNaC::is_a(), GiNaC::ex::nops(), GiNaC::ex::op(), GiNaC::basic::setflag(), and x.

Referenced by eval_integ().

8.77.3.11 get_free_indices()

```
exvector GiNaC::integral::get_free_indices () const  [override], [virtual]
```

Return a vector containing the free indices of an expression.

Reimplemented from GiNaC::basic.

References a, b, f, and GiNaC::ex::get_free_indices().

8.77.3.12 return_type()

```
unsigned GiNaC::integral::return_type () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References f, and GiNaC::ex::return_type().

8.77.3.13 return_type_tinfo()

```
return_type_t GiNaC::integral::return_type_tinfo () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References f, and GiNaC::ex::return_type_tinfo().

8.77.3.14 conjugate()

```
ex GiNaC::integral::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References a, GiNaC::are_ex_trivially_equal(), b, GiNaC::ex::conjugate(), GiNaC::dynallocate(), GiNaC::basic::ex, f, GiNaC::ex::subs(), and x.

8.77.3.15 eval_integ()

```
ex GiNaC::integral::eval_integ () const  [override], [virtual]
```

Evaluate integrals, if result is known.

Reimplemented from GiNaC::basic.

References a, b, GiNaC::ex::eval_integ(), GiNaC::basic::ex, expand(), GiNaC::status_flags::expanded, f, GiNaC::basic::flags, GiNaC::ex::has(), GiNaC::is_a(), GiNaC::log(), GiNaC::ex::op(), GiNaC::power::power(), GiNaC::ex::subs(), and x.

8.77.3.16 archive()

```
void GiNaC::integral::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References a, GiNaC::archive_node::add_ex(), b, f, and x.

8.77.3.17 read_archive()

```
void GiNaC::integral::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References a, b, f, GiNaC::archive_node::find_ex(), and x.

8.77.3.18 derivative()

```
ex GiNaC::integral::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Default implementation of ex::diff().

It maps the operation on the operands (or returns 0 when the object has no operands).

See also

ex::diff

Reimplemented from GiNaC::basic.

References a, b, GiNaC::ex::diff(), GiNaC::basic::ex, f, integral(), GiNaC::ex::subs(), and x.

8.77.3.19 series()

```
ex GiNaC::integral::series (
            const relational & r,
            int order,
            unsigned options = 0) const  [override], [protected], [virtual]
```

Default implementation of ex::series().

This performs Taylor expansion.

See also

ex::series

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, a, b, GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), f, integral(), GiNaC::is_order_function(), GiNaC::ex::nops(), GiNaC::ex::series(), GiNaC::ex::subs(), and x.

8.77.3.20 do_print()

```
void GiNaC::integral::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References a, b, f, GiNaC::ex::print(), GiNaC::print_context::s, and x.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.77.3.21 do_print_latex()

```
void GiNaC::integral::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

References a, b, GiNaC::ex_to(), f, precedence(), GiNaC::ex::print(), GiNaC::print_context::s, and x.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.77.4 Member Data Documentation

8.77.4.1 max_integration_level

```
int GiNaC::integral::max_integration_level = 15  [static]
```

Referenced by GiNaC::adaptivesimpson().

8.77.4.2 relative_integration_error

```
ex GiNaC::integral::relative_integration_error = 1e-8  [static]
```

Referenced by GiNaC::GINAC_DECLARE_UNARCHIVER().

8.77.4.3 x

```
ex GiNaC::integral::x  [private]
```

Referenced by archive(), conjugate(), derivative(), do_print(), do_print_latex(), eval(), eval_integ(), evalf(), expand(), integral(), let_op(), op(), read_archive(), and series().

8.77.4.4 a

```
ex GiNaC::integral::a  [private]
```

Referenced by archive(), conjugate(), degree(), derivative(), do_print(), do_print_latex(), eval(), eval_integ(), evalf(), expand(), get_free_indices(), integral(), ldegree(), let_op(), op(), read_archive(), and series().

8.77.4.5 b

```
ex GiNaC::integral::b  [private]
```

Referenced by archive(), conjugate(), degree(), derivative(), do_print(), do_print_latex(), eval(), eval_integ(), evalf(), expand(), get_free_indices(), integral(), ldegree(), let_op(), op(), read_archive(), and series().

8.77.4.6 f

```
ex GiNaC::integral::f  [private]
```

Referenced by archive(), conjugate(), degree(), derivative(), do_print(), do_print_latex(), eval(), eval_integ(), eval_ncmul(), evalf(), expand(), get_free_indices(), integral(), ldegree(), let_op(), op(), read_archive(), return_type(), return_type_tinfo(), and series().

The documentation for this class was generated from the following files:

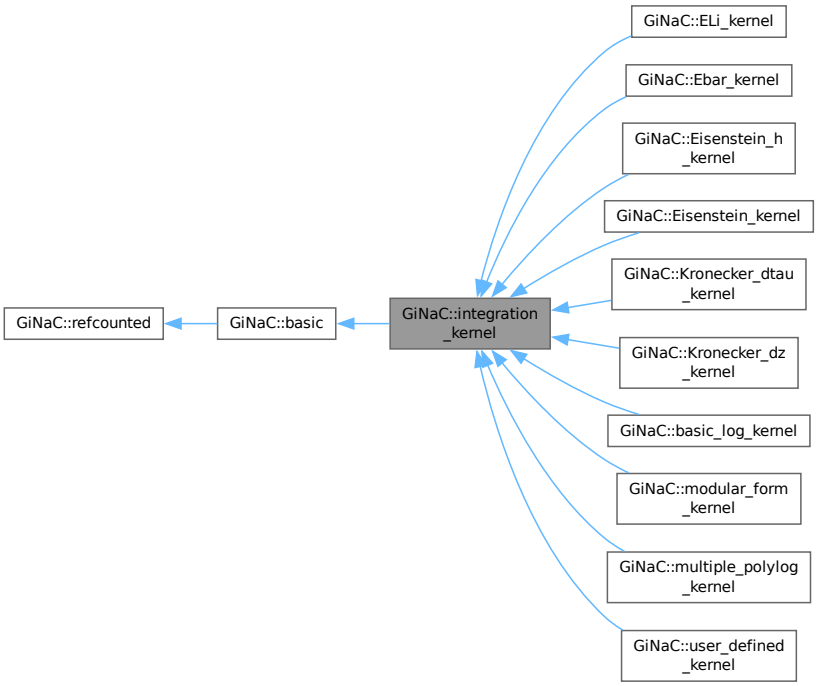
- integral.h
- indexed.cpp
- integral.cpp
- pseries.cpp

8.78 GiNaC::integration_kernel Class Reference

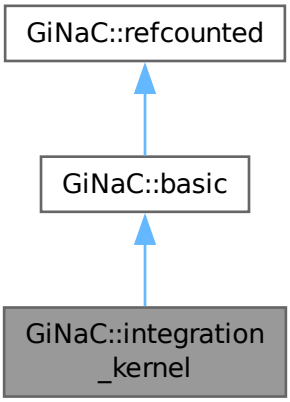
The base class for integration kernels for iterated integrals.

```
#include <integration_kernel.h>
```

Inheritance diagram for GiNaC::integration_kernel:



Collaboration diagram for GiNaC::integration_kernel:



Public Member Functions

- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- virtual bool is_numeric (void) const

  This routine returns true, if the integration kernel can be evaluated numerically.
- virtual ex Laurent_series (const ex &x, int order) const

  Returns the Laurent series, starting possibly with the pole term.
- virtual ex get_numerical_value (const ex &lambda, int N_trunc=0) const

  Evaluates the integrand at lambda.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

Create a clone of this object on the heap.

- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

Sort expanded expression in terms of powers of some object(s).

- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- virtual bool uses_Laurent_series () const

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- virtual cln::cl_N series_coeff_impl (int i) const

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.78.1 Detailed Description

The base class for integration kernels for iterated integrals.

This class represents the differential one-form

$$\omega = d\lambda$$

The integration variable is a dummy variable and does not need to be specified.

8.78.2 Member Function Documentation

8.78.2.1 series()

```
ex GiNaC::integration_kernel::series (
             const relational & r,
             int order,
             unsigned options = 0) const  [override], [virtual]
```

Default implementation of ex::series().

This performs Taylor expansion.

See also

ex::series

Reimplemented from GiNaC::basic.

Reimplemented in GiNaC::modular_form_kernel.

References GiNaC::basic::ex, Laurent_series(), GiNaC::relational::lhs(), GiNaC::relational::rhs(), and series().

Referenced by series().

8.78.2.2 has_trailing_zero()

```
bool GiNaC::integration_kernel::has_trailing_zero (
             void ) const  [virtual]
```

This routine returns true, if the integration kernel has a trailing zero.

References has_trailing_zero(), and series_coeff().

Referenced by has_trailing_zero().

8.78.2.3 is_numeric()

```
bool GiNaC::integration_kernel::is_numeric (
            void ) const  [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented in GiNaC::Ebar_kernel, GiNaC::Eisenstein_h_kernel, GiNaC::Eisenstein_kernel, GiNaC::ELi_kernel, GiNaC::Kronecker_dtau_kernel, GiNaC::Kronecker_dz_kernel, GiNaC::modular_form_kernel, GiNaC::multiple_polylog_kernel, and GiNaC::user_defined_kernel.

References is_numeric().

Referenced by is_numeric().

8.78.2.4 Laurent_series()

```
ex GiNaC::integration_kernel::Laurent_series (
            const ex & x,
            int order) const  [virtual]
```

Returns the Laurent series, starting possibly with the pole term.

Neglected terms are of order $O(x^o rder)$.

Reimplemented in GiNaC::Eisenstein_h_kernel, GiNaC::Eisenstein_kernel, GiNaC::modular_form_kernel, and GiNaC::user_defined_kernel.

References GiNaC::basic::ex, Laurent_series(), GiNaC::pow(), GiNaC::ex::series(), and series_coeff().

Referenced by Laurent_series(), series(), and series_coeff().

8.78.2.5 get_numerical_value()

```
ex GiNaC::integration_kernel::get_numerical_value (
            const ex & lambda,
            int N_trunc = 0) const  [virtual]
```

Evaluates the integrand at lambda.

Reimplemented in GiNaC::Ebar_kernel, GiNaC::Eisenstein_h_kernel, GiNaC::Eisenstein_kernel, GiNaC::ELi_kernel, GiNaC::Kronecker_dtau_kernel, GiNaC::Kronecker_dz_kernel, and GiNaC::modular_form_kernel.

References GiNaC::basic::ex, get_numerical_value(), and get_numerical_value_impl().

Referenced by get_numerical_value().

8.78.2.6 uses_Laurent_series()

```
bool GiNaC::integration_kernel::uses_Laurent_series () const  [protected], [virtual]
```

Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).

Returns false if this is not the case (and the class has an implementation of series_coeff_impl).

Reimplemented in GiNaC::Eisenstein_h_kernel, GiNaC::Eisenstein_kernel, GiNaC::modular_form_kernel, and GiNaC::user_defined_kernel.

References uses_Laurent_series().

Referenced by series_coeff(), and uses_Laurent_series().

8.78.2.7 series_coeff_impl()

```
cln::cl_N GiNaC::integration_kernel::series_coeff_impl (
            int i) const  [protected], [virtual]
```

For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.

The i-th coefficient corresponds to the power λ^{i-1}.

Reimplemented in GiNaC::basic_log_kernel, GiNaC::Ebar_kernel, GiNaC::ELi_kernel, GiNaC::Kronecker_dtau_kernel, GiNaC::Kronecker_dz_kernel, and GiNaC::multiple_polylog_kernel.

References series_coeff_impl().

Referenced by series_coeff(), and series_coeff_impl().

8.78.2.8 get_cache_size()

```
size_t GiNaC::integration_kernel::get_cache_size (
            void ) const
```

Returns the current size of the cache.

References get_cache_size(), and series_vec.

Referenced by get_cache_size().

8.78.2.9 set_cache_step()

```
void GiNaC::integration_kernel::set_cache_step (
            int cache_steps) const
```

Sets the step size by which the cache is increased.

References cache_step_size, and set_cache_step().

Referenced by set_cache_step().

8.78.2.10 get_series_coeff()

```
ex GiNaC::integration_kernel::get_series_coeff (
            int i) const
```

Wrapper around series_coeff(i), converts cl_N to numeric.

References GiNaC::basic::ex, get_series_coeff(), and series_coeff().

Referenced by get_series_coeff().

8.78.2.11 series_coeff()

```
cln::cl_N GiNaC::integration_kernel::series_coeff (
            int i) const
```

Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_↩ series.

The method series_coeff_impl can be used, if a single coefficient can be computed independently of the others.

The method Laurent_series can be used, if it is more efficient to compute a Laurent series in one shot and to determine a range of coefficients from this Laurent series.

References cache_step_size, GiNaC::ex::coeff(), GiNaC::ex::evalf(), GiNaC::basic::ex, GiNaC::ex_to(), Laurent_series(), series_coeff(), series_coeff_impl(), series_vec, and uses_Laurent_series().

Referenced by get_numerical_value_impl(), get_series_coeff(), has_trailing_zero(), Laurent_series(), and series_coeff().

8.78.2.12 get_numerical_value_impl()

```
ex GiNaC::integration_kernel::get_numerical_value_impl (
            const ex & lambda,
            const ex & pre,
            int shift,
            int N_trunc) const  [protected]
```

The actual implementation for computing a numerical value for the integrand.

References GiNaC::Digits, GiNaC::ex::evalf(), GiNaC::basic::ex, GiNaC::ex_to(), get_numerical_value_impl(), and series_coeff().

Referenced by GiNaC::Ebar_kernel::get_numerical_value(), GiNaC::Eisenstein_h_kernel::get_numerical_value(), GiNaC::Eisenstein_kernel::get_numerical_value(), GiNaC::ELi_kernel::get_numerical_value(), get_numerical_value(), GiNaC::Kronecker_dtau_kernel::get_numerical_value(), GiNaC::Kronecker_dz_kernel::get_numerical_value(), GiNaC::modular_form_kernel::get_numerical_value(), and get_numerical_value_impl().

8.78.2.13 do_print()

```
void GiNaC::integration_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References do_print(), and GiNaC::print_context::s.

Referenced by do_print(), and GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.78.3 Member Data Documentation

8.78.3.1 cache_step_size

```
int GiNaC::integration_kernel::cache_step_size  [mutable], [protected]
```

Referenced by series_coeff(), and set_cache_step().

8.78.3.2 series_vec

```
std::vector<cln::cl_N> GiNaC::integration_kernel::series_vec  [mutable], [protected]
```

Referenced by get_cache_size(), and series_coeff().

The documentation for this class was generated from the following files:

- integration_kernel.h
- integration_kernel.cpp

8.79 GiNaC::is_not_a_clifford Struct Reference

Predicate for finding non-clifford objects.

Public Member Functions

- bool operator() (const ex &e)

8.79.1 Detailed Description

Predicate for finding non-clifford objects.

8.79.2 Member Function Documentation

8.79.2.1 operator()()

```
bool GiNaC::is_not_a_clifford::operator() (
            const ex & e)  [inline]
```

References GiNaC::is_a().

The documentation for this struct was generated from the following file:

- clifford.cpp

8.80 GiNaC::is_summation_idx Struct Reference

Public Member Functions

- bool operator() (const ex &e)

8.80.1 Member Function Documentation

8.80.1.1 operator()()

```
bool GiNaC::is_summation_idx::operator() (
            const ex & e)  [inline]
```

References GiNaC::is_dummy_pair().

The documentation for this struct was generated from the following file:

- indexed.cpp

8.81 GiNaC::iterated_integral2_SERIAL Class Reference

Complete elliptic integral of the first kind.

```
#include <inifcns.h>
```

Static Public Attributes

- static unsigned serial

8.81.1 Detailed Description

Complete elliptic integral of the first kind.

Complete elliptic integral of the second kind. Iterated integral.

8.81.2 Member Data Documentation

8.81.2.1 serial

```
unsigned GiNaC::iterated_integral2_SERIAL::serial  [static]
```

Initial value:

```
                                    =
function::register_new(function_options("iterated_integral", 2).
                       eval_func(iterated_integral2_eval).
                       evalf_func(iterated_integral2_evalf).
                           do_not_evalf_params().
                       overloaded(2))
```

Referenced by GiNaC::iterated_integral().

The documentation for this class was generated from the following files:

- inifcns.h
- inifcns_elliptic.cpp

8.82 GiNaC::iterated_integral3_SERIAL Class Reference

Iterated integral with explicit truncation.

`#include <inifcns.h>`

Static Public Attributes

- static unsigned serial

8.82.1 Detailed Description

Iterated integral with explicit truncation.

8.82.2 Member Data Documentation

8.82.2.1 serial

`unsigned GiNaC::iterated_integral3_SERIAL::serial  [static]`

Initial value:

```
                                        =
function::register_new(function_options("iterated_integral", 3).
                       eval_func(iterated_integral3_eval).
                       evalf_func(iterated_integral3_evalf).
                           do_not_evalf_params().
                       overloaded(2))
```

Referenced by GiNaC::iterated_integral().

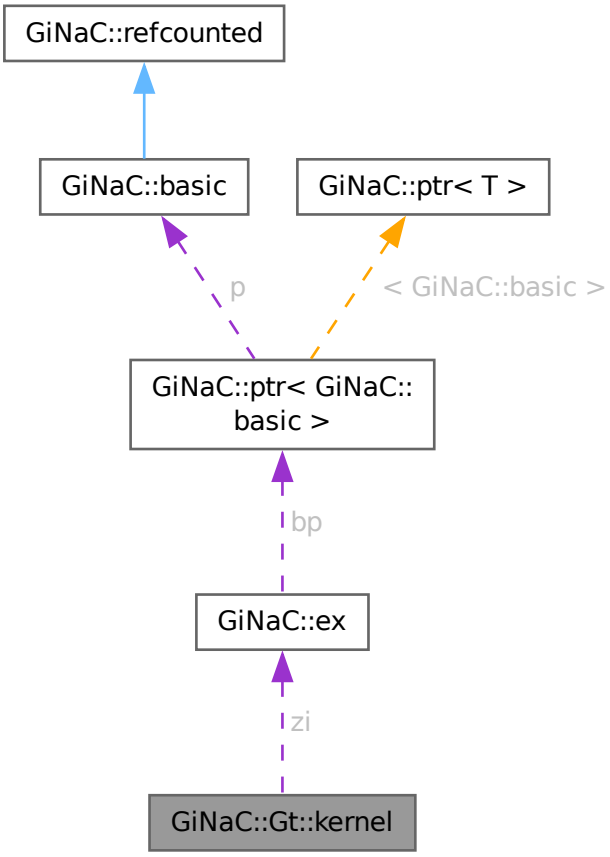
The documentation for this class was generated from the following files:

- inifcns.h
- inifcns_elliptic.cpp

8.83 GiNaC::Gt::kernel Struct Reference

`#include <Gt.h>`

Collaboration diagram for GiNaC::Gt::kernel:



Public Member Functions

- kernel ()
- kernel (size_t ni, ex zi, int deform=0)
- kernel (lst arg)

Public Attributes

- size_t ni
- ex zi
- int deform

Private Member Functions

- void check ()

Friends

- class Gt

8.83.1 Constructor & Destructor Documentation

8.83.1.1 kernel() [1/3]

```
GiNaC::Gt::kernel::kernel ()
```

References deform, GiNaC::basic::ex, ni, and zi.

Referenced by kernel().

8.83.1.2 kernel() [2/3]

```
GiNaC::Gt::kernel::kernel (
            size_t ni,
            ex zi,
            int deform = 0)
```

References check(), deform, GiNaC::basic::ex, kernel(), ni, and zi.

8.83.1.3 kernel() [3/3]

```
GiNaC::Gt::kernel::kernel (
            lst arg)
```

References check(), deform, ni, GiNaC::container< C >::nops(), GiNaC::container< C >::op(), GiNaC::Gt::to_integer(), and zi.

8.83.2 Member Function Documentation

8.83.2.1 check()

```
void GiNaC::Gt::kernel::check ()  [private]
```

References deform, ni, and zi.

Referenced by kernel(), and kernel().

8.83.3 Friends And Related Symbol Documentation

8.83.3.1 Gt

```
friend class Gt  [friend]
```

References Gt.

Referenced by Gt.

8.83.4 Member Data Documentation

8.83.4.1 ni

```
size_t GiNaC::Gt::kernel::ni
```

Referenced by check(), kernel(), kernel(), kernel(), and GiNaC::Gt::tauToFundamental().

8.83.4.2 zi

```
ex GiNaC::Gt::kernel::zi
```

Referenced by check(), kernel(), kernel(), and kernel().

8.83.4.3 deform

```
int GiNaC::Gt::kernel::deform
```

Referenced by check(), kernel(), kernel(), and kernel().

The documentation for this struct was generated from the following files:

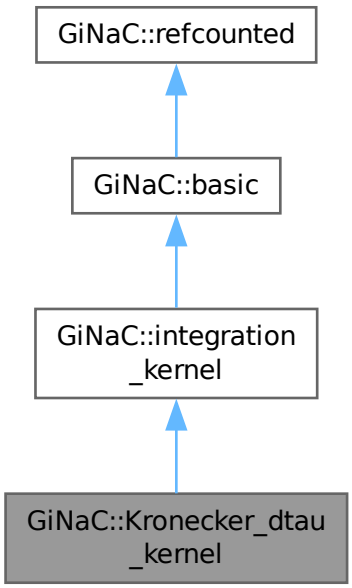
- Gt.h
- Gt.cpp

8.84 GiNaC::Kronecker_dtau_kernel Class Reference

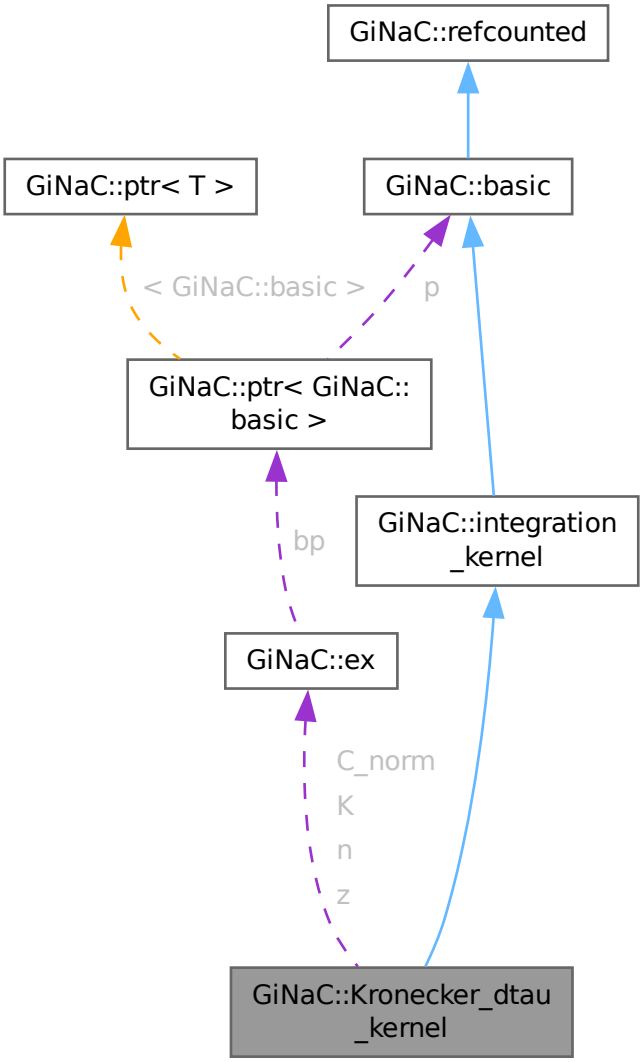
The kernel corresponding to integrating the Kronecker coefficient function $g^{(n)}(z_j, K\tau)$ in τ (or equivalently in $\bar{q}$).

```
#include <integration_kernel.h>
```

Inheritance diagram for GiNaC::Kronecker_dtau_kernel:



Collaboration diagram for GiNaC::Kronecker_dtau_kernel:



Public Member Functions

- Kronecker_dtau_kernel (const ex &n, const ex &z, const ex &K=numeric(1), const ex &C_norm=numeric(1))
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- bool is_numeric (void) const override

  This routine returns true, if the integration kernel can be evaluated numerically.
- ex get_numerical_value (const ex &qbar, int N_trunc=0) const override

  Returns the value of the g^(n)(z,K∗tau), where tau is given by qbar.

Public Member Functions inherited from GiNaC::integration_kernel

- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- virtual ex Laurent_series (const ex &x, int order) const

  Returns the Laurent series, starting possibly with the pole term.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

Load (deserialize) the object from an archive node.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- cln::cl_N series_coeff_impl (int i) const override

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual bool uses_Laurent_series () const

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex n
- ex z
- ex K
- ex C_norm

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.84.1 Detailed Description

The kernel corresponding to integrating the Kronecker coefficient function $g^{(n)}(z_j, K\tau)$ in τ (or equivalently in $\bar{q}$).

This class represents the differential one-form

$$\omega_{n,K}^{\mathrm{Kronecker},\tau}(z_j) = \frac{C_n K(n-1)}{(2\pi i)^n} g^{(n)}(z_j, K\tau)\frac{d\bar{q}}{\bar{q}}$$

8.84.2 Constructor & Destructor Documentation

8.84.2.1 Kronecker_dtau_kernel()

```
GiNaC::Kronecker_dtau_kernel::Kronecker_dtau_kernel (
             const ex & n,
             const ex & z,
             const ex & K = numeric(1),
             const ex & C_norm = numeric(1))
```

References C_norm, K, Kronecker_dtau_kernel(), n, and z.

Referenced by Kronecker_dtau_kernel().

8.84.3 Member Function Documentation

8.84.3.1 nops()

```
size_t GiNaC::Kronecker_dtau_kernel::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.84.3.2 op()

```
ex GiNaC::Kronecker_dtau_kernel::op (
             size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References C_norm, K, n, and z.

8.84.3.3 let_op()

```
ex & GiNaC::Kronecker_dtau_kernel::let_op (
             size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References C_norm, GiNaC::basic::ensure_if_modifiable(), K, n, and z.

8.84.3.4 is_numeric()

```
bool GiNaC::Kronecker_dtau_kernel::is_numeric (
            void ) const  [override], [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented from GiNaC::integration_kernel.

References C_norm, GiNaC::ex::evalf(), GiNaC::ex::info(), K, n, GiNaC::info_flags::nonnegint, GiNaC::info_flags::numeric, GiNaC::info_flags::posint, and z.

8.84.3.5 get_numerical_value()

```
ex GiNaC::Kronecker_dtau_kernel::get_numerical_value (
            const ex & qbar,
            int N_trunc = 0) const  [override], [virtual]
```

Returns the value of the g^(n)(z,K∗tau), where tau is given by qbar.

Reimplemented from GiNaC::integration_kernel.

References C_norm, GiNaC::Ebar_kernel::Ebar_kernel(), GiNaC::basic::evalf(), GiNaC::ex::evalf(), GiNaC::ex_to(), GiNaC::exp(), GiNaC::integration_kernel::get_numerical_value_impl(), GiNaC::I, K, n, GiNaC::numeric::numeric(), GiNaC::Pi, GiNaC::pow(), and z.

8.84.3.6 series_coeff_impl()

```
cln::cl_N GiNaC::Kronecker_dtau_kernel::series_coeff_impl (
            int i) const  [override], [protected], [virtual]
```

For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.

The i-th coefficient corresponds to the power λ^{i-1}.

Reimplemented from GiNaC::integration_kernel.

References GiNaC::bernoulli(), C_norm, GiNaC::basic::evalf(), GiNaC::ex::evalf(), GiNaC::ex_to(), GiNaC::exp(), GiNaC::factorial(), GiNaC::I, K, n, GiNaC::Pi, GiNaC::numeric::to_cl_N(), GiNaC::numeric::to_int(), and z.

8.84.3.7 do_print()

```
void GiNaC::Kronecker_dtau_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References C_norm, K, n, GiNaC::ex::print(), GiNaC::print_context::s, and z.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.84.4 Member Data Documentation

8.84.4.1 n

ex GiNaC::Kronecker_dtau_kernel::n [protected]

Referenced by do_print(), get_numerical_value(), is_numeric(), Kronecker_dtau_kernel(), let_op(), op(), and series_coeff_impl().

8.84.4.2 z

ex GiNaC::Kronecker_dtau_kernel::z [protected]

Referenced by do_print(), get_numerical_value(), is_numeric(), Kronecker_dtau_kernel(), let_op(), op(), and series_coeff_impl().

8.84.4.3 K

ex GiNaC::Kronecker_dtau_kernel::K [protected]

Referenced by do_print(), get_numerical_value(), is_numeric(), Kronecker_dtau_kernel(), let_op(), op(), and series_coeff_impl().

8.84.4.4 C_norm

ex GiNaC::Kronecker_dtau_kernel::C_norm [protected]

Referenced by do_print(), get_numerical_value(), is_numeric(), Kronecker_dtau_kernel(), let_op(), op(), and series_coeff_impl().

The documentation for this class was generated from the following files:

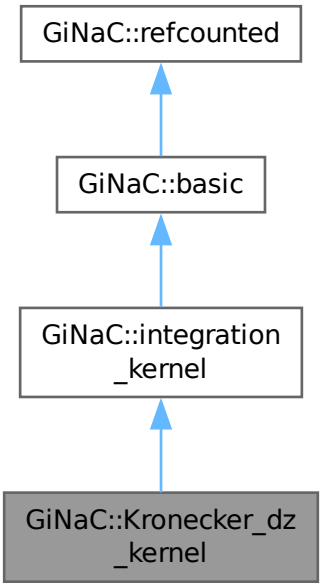
- integration_kernel.h
- integration_kernel.cpp

8.85 GiNaC::Kronecker_dz_kernel Class Reference

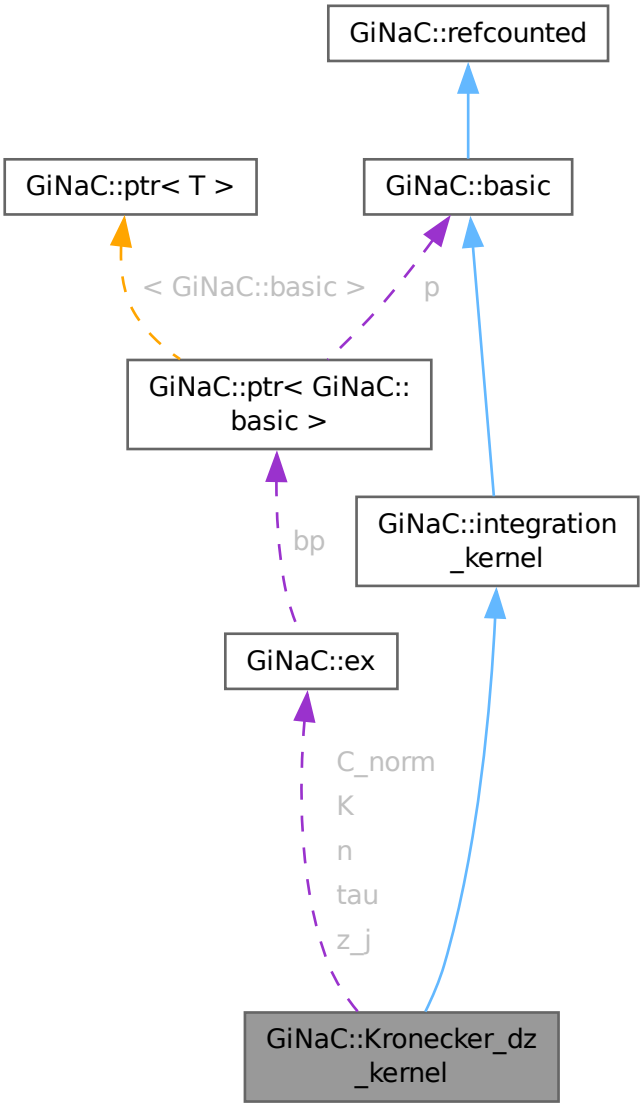
The kernel corresponding to integrating the Kronecker coefficient function $g^{(n-1)}(z - z_j, K\tau)$ in z.

`#include <integration_kernel.h>`

Inheritance diagram for GiNaC::Kronecker_dz_kernel:



Collaboration diagram for GiNaC::Kronecker_dz_kernel:



Public Member Functions

- Kronecker_dz_kernel (const ex &n, const ex &z_j, const ex &tau, const ex &K=numeric(1), const ex &C_↩ norm=numeric(1))
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- bool is_numeric (void) const override

This routine returns true, if the integration kernel can be evaluated numerically.

- ex get_numerical_value (const ex &z, int N_trunc=0) const override

  Returns the value of the g^(n-1)(z-z_j,K∗tau).

Public Member Functions inherited from GiNaC::integration_kernel

- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- virtual ex Laurent_series (const ex &x, int order) const

  Returns the Laurent series, starting possibly with the pole term.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

Information about the object.

- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.

- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- cln::cl_N series_coeff_impl (int i) const override

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual bool uses_Laurent_series () const

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex n
- ex z_j
- ex tau
- ex K
- ex C_norm

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.85.1 Detailed Description

The kernel corresponding to integrating the Kronecker coefficient function $g^{(n-1)}(z - z_j, K\tau)$ in z.

This class represents the differential one-form

$$\omega_{n,K}^{\mathrm{Kronecker},z}(z_j, \tau) = C_n (2\pi i)^{2-n} g^{(n-1)}(z - z_j, K\tau) dz$$

8.85.2 Constructor & Destructor Documentation

8.85.2.1 Kronecker_dz_kernel()

```
GiNaC::Kronecker_dz_kernel::Kronecker_dz_kernel (
              const ex & n,
              const ex & z_j,
              const ex & tau,
              const ex & K = numeric(1),
              const ex & C_norm = numeric(1))
```

References C_norm, K, Kronecker_dz_kernel(), n, tau, and z_j.

Referenced by Kronecker_dz_kernel().

8.85.3 Member Function Documentation

8.85.3.1 nops()

```
size_t GiNaC::Kronecker_dz_kernel::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.85.3.2 op()

```
ex GiNaC::Kronecker_dz_kernel::op (
              size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References C_norm, K, n, tau, and z_j.

8.85.3.3 let_op()

```
ex & GiNaC::Kronecker_dz_kernel::let_op (
              size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References C_norm, GiNaC::basic::ensure_if_modifiable(), K, n, tau, and z_j.

8.85.3.4 is_numeric()

```
bool GiNaC::Kronecker_dz_kernel::is_numeric (
            void ) const  [override], [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented from GiNaC::integration_kernel.

References C_norm, GiNaC::ex::evalf(), GiNaC::ex::info(), K, n, GiNaC::info_flags::nonnegint, GiNaC::info_flags::numeric, GiNaC::info_flags::posint, tau, and z_j.

8.85.3.5 get_numerical_value()

```
ex GiNaC::Kronecker_dz_kernel::get_numerical_value (
            const ex & z,
            int N_trunc = 0) const  [override], [virtual]
```

Returns the value of the g^(n-1)(z-z_j,K∗tau).

Reimplemented from GiNaC::integration_kernel.

References C_norm, GiNaC::ex_to(), GiNaC::integration_kernel::get_numerical_value_impl(), GiNaC::I, n, GiNaC::Pi, and GiNaC::pow().

8.85.3.6 series_coeff_impl()

```
cln::cl_N GiNaC::Kronecker_dz_kernel::series_coeff_impl (
            int i) const  [override], [protected], [virtual]
```

For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.

The i-th coefficient corresponds to the power λ^{i-1}.

Reimplemented from GiNaC::integration_kernel.

References GiNaC::bernoulli(), C_norm, GiNaC::Ebar_kernel::Ebar_kernel(), GiNaC::basic::evalf(), GiNaC::ex::evalf(), GiNaC::ex_to(), GiNaC::exp(), GiNaC::factorial(), GiNaC::Ebar_kernel::get_numerical_value(), GiNaC::I, GiNaC::is_even(), K, n, GiNaC::numeric::numeric(), GiNaC::Pi, GiNaC::pow(), tau, and z_j.

8.85.3.7 do_print()

```
void GiNaC::Kronecker_dz_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References C_norm, K, n, GiNaC::ex::print(), GiNaC::print_context::s, tau, and z_j.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.85.4 Member Data Documentation

8.85.4.1 n

```
ex GiNaC::Kronecker_dz_kernel::n  [protected]
```

Referenced by do_print(), get_numerical_value(), is_numeric(), Kronecker_dz_kernel(), let_op(), op(), and series_coeff_impl().

8.85.4.2 z_j

```
ex GiNaC::Kronecker_dz_kernel::z_j  [protected]
```

Referenced by do_print(), is_numeric(), Kronecker_dz_kernel(), let_op(), op(), and series_coeff_impl().

8.85.4.3 tau

```
ex GiNaC::Kronecker_dz_kernel::tau  [protected]
```

Referenced by do_print(), is_numeric(), Kronecker_dz_kernel(), let_op(), op(), and series_coeff_impl().

8.85.4.4 K

```
ex GiNaC::Kronecker_dz_kernel::K  [protected]
```

Referenced by do_print(), is_numeric(), Kronecker_dz_kernel(), let_op(), op(), and series_coeff_impl().

8.85.4.5 C_norm

```
ex GiNaC::Kronecker_dz_kernel::C_norm  [protected]
```

Referenced by do_print(), get_numerical_value(), is_numeric(), Kronecker_dz_kernel(), let_op(), op(), and series_coeff_impl().

The documentation for this class was generated from the following files:

- integration_kernel.h
- integration_kernel.cpp

8.86 GiNaC::lanczos_coeffs Class Reference

Public Member Functions

- lanczos_coeffs ()
- bool sufficiently_accurate (int digits)
- int get_order () const
- cln::cl_N calc_lanczos_A (const cln::cl_N &) const

Private Attributes

- std::vector< cln::cl_N > * current_vector

Static Private Attributes

- static std::vector< cln::cl_N > * coeffs = nullptr

8.86.1 Constructor & Destructor Documentation

8.86.1.1 lanczos_coeffs()

```
GiNaC::lanczos_coeffs::lanczos_coeffs ()
```

References coeffs.

8.86.2 Member Function Documentation

8.86.2.1 sufficiently_accurate()

```
bool GiNaC::lanczos_coeffs::sufficiently_accurate (
            int digits)
```

References coeffs, and current_vector.

Referenced by GiNaC::lgamma(), and GiNaC::tgamma().

8.86.2.2 get_order()

```
int GiNaC::lanczos_coeffs::get_order () const  [inline]
```

References current_vector.

Referenced by GiNaC::lgamma(), and GiNaC::tgamma().

8.86.2.3 calc_lanczos_A()

```
cln::cl_N GiNaC::lanczos_coeffs::calc_lanczos_A (
            const cln::cl_N & x) const
```

References current_vector.

Referenced by GiNaC::lgamma(), and GiNaC::tgamma().

8.86.3 Member Data Documentation

8.86.3.1 coeffs

```
std::vector< cln::cl_N > * GiNaC::lanczos_coeffs::coeffs = nullptr  [static], [private]
```

Referenced by lanczos_coeffs(), and sufficiently_accurate().

8.86.3.2 current_vector

```
std::vector<cln::cl_N>* GiNaC::lanczos_coeffs::current_vector  [private]
```

Referenced by calc_lanczos_A(), get_order(), and sufficiently_accurate().

The documentation for this class was generated from the following file:

- numeric.cpp

8.87 std::less< GiNaC::ptr< T > > Struct Template Reference

Specialization of std::less for ptr<T> to enable ordering of ptr<T> objects (e.g.

```
#include <ptr.h>
```

Public Member Functions

- bool operator() (const GiNaC::ptr< T > &lhs, const GiNaC::ptr< T > &rhs) const

8.87.1 Detailed Description

template<class T>
struct std::less< GiNaC::ptr< T > >

Specialization of std::less for ptr<T> to enable ordering of ptr<T> objects (e.g.

for the use as std::map keys).

8.87.2 Member Function Documentation

8.87.2.1 operator()()

```
template<class T>
bool std::less< GiNaC::ptr< T > >::operator() (
            const GiNaC::ptr< T > & lhs,
            const GiNaC::ptr< T > & rhs) const  [inline]
```

The documentation for this struct was generated from the following file:

- ptr.h

8.88 GiNaC::library_init Class Reference

Helper class to initialize the library.

`#include <ex.h>`

Public Member Functions

- library_init ()
  Ctor of static initialization helpers.
- ~library_init ()
  Dtor of static initialization helpers.

Static Private Member Functions

- static void init_unarchivers ()

Static Private Attributes

- static int count = 0
  How many static objects were created?

8.88.1 Detailed Description

Helper class to initialize the library.

There must be one static object of this class in every object file that makes use of our flyweights in order to guarantee proper initialization. Hence we put it into this file which is included by every relevant file anyways. This is modeled after section [ios::Init] of the C++ standard, where cout and friends are set up.

See also

utils.cpp

8.88.2 Constructor & Destructor Documentation

8.88.2.1 library_init()

```
GiNaC::library_init::library_init ()
```

Ctor of static initialization helpers.

The fist call to this is going to initialize the library, the others do nothing.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::_ex10, GiNaC::_ex11, GiNaC::_ex12, GiNaC::_ex120, GiNaC::_ex15, GiNaC::_ex18, GiNaC::_ex1_2, GiNaC::_ex1_3, GiNaC::_ex1_4, GiNaC::_ex2, GiNaC::_ex20, GiNaC::_ex24, GiNaC::_ex25, GiNaC::_ex3, GiNaC::_ex30, GiNaC::_ex4, GiNaC::_ex48, GiNaC::_ex5, GiNaC::_ex6, GiNaC::_ex60, GiNaC::_ex7, GiNaC::_ex8, GiNaC::_ex9, GiNaC::_ex_1, GiNaC::_ex_10, GiNaC::_ex_11, GiNaC::_ex_12, GiNaC::_ex_120, GiNaC::_ex_15, GiNaC::_ex_18, GiNaC::_ex_1_2, GiNaC::_ex_1_3, GiNaC::_ex_1_4, GiNaC::_ex_2, GiNaC::_ex_20, GiNaC::_ex_24, GiNaC::_ex_25, GiNaC::_ex_3, GiNaC::_ex_30, GiNaC::_ex_4, GiNaC::_ex_48, GiNaC::_ex_5, GiNaC::_ex_6, GiNaC::_ex_60, GiNaC::_ex_7, GiNaC::_ex_8, GiNaC::_ex_9, GiNaC::_num0_bp, GiNaC::_num0_p, GiNaC::_num10_p, GiNaC::_num11_p, GiNaC::_num120_p, GiNaC::_num12_p, GiNaC::_num15_p, GiNaC::_num18_p, GiNaC::_num1_2_p, GiNaC::_num1_3_p, GiNaC::_num1_4_p, GiNaC::_num1_p, GiNaC::_num20_p, GiNaC::_num24_p, GiNaC::_num25_p, GiNaC::_num2_p, GiNaC::_num30_p, GiNaC::_num3_p, GiNaC::_num48_p, GiNaC::_num4_p, GiNaC::_num5_p, GiNaC::_num60_p, GiNaC::_num6_p, GiNaC::_num7_p, GiNaC::_num8_p, GiNaC::_num9_p, GiNaC::_num_10_p, GiNaC::_num_11_p, GiNaC::_num_120_p, GiNaC::_num_12_p, GiNaC::_num_15_p, GiNaC::_num_18_p, GiNaC::_num_1_2_p, GiNaC::_num_1_3_p, GiNaC::_num_1_4_p, GiNaC::_num_1_p, GiNaC::_num_20_p, GiNaC::_num_24_p, GiNaC::_num_25_p, GiNaC::_num_2_p, GiNaC::_num_30_p, GiNaC::_num_3_p, GiNaC::_num_48_p, GiNaC::_num_4_p, GiNaC::_num_5_p, GiNaC::_num_60_p, GiNaC::_num_6_p, GiNaC::_num_7_p, GiNaC::_num_8_p, GiNaC::_num_9_p, count, GiNaC::dynallocate(), and GiNaC::ex::ex().

8.88.2.2 ~library_init()

```
GiNaC::library_init::~library_init ()
```

Dtor of static initialization helpers.

The last call to this is going to shut down the library, the others do nothing.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::_ex10, GiNaC::_ex11, GiNaC::_ex12, GiNaC::_ex120, GiNaC::_ex15, GiNaC::_ex18, GiNaC::_ex1_2, GiNaC::_ex1_3, GiNaC::_ex1_4, GiNaC::_ex2, GiNaC::_ex20, GiNaC::_ex24, GiNaC::_ex25, GiNaC::_ex3, GiNaC::_ex30, GiNaC::_ex4, GiNaC::_ex48, GiNaC::_ex5, GiNaC::_ex6, GiNaC::_ex60, GiNaC::_ex7, GiNaC::_ex8, GiNaC::_ex9, GiNaC::_ex_1, GiNaC::_ex_10, GiNaC::_ex_11, GiNaC::_ex_12, GiNaC::_ex_120, GiNaC::_ex_15, GiNaC::_ex_18, GiNaC::_ex_1_2, GiNaC::_ex_1_3, GiNaC::_ex_1_4, GiNaC::_ex_2, GiNaC::_ex_20, GiNaC::_ex_24, GiNaC::_ex_25, GiNaC::_ex_3, GiNaC::_ex_30, GiNaC::_ex_4, GiNaC::_ex_48, GiNaC::_ex_5, GiNaC::_ex_6, GiNaC::_ex_60, GiNaC::_ex_7, GiNaC::_ex_8, GiNaC::_ex_9, and count.

8.88.3 Member Function Documentation

8.88.3.1 init_unarchivers()

```
void GiNaC::library_init::init_unarchivers ()  [static], [private]
```

8.88.4 Member Data Documentation

8.88.4.1 count

```
int GiNaC::library_init::count = 0  [static], [private]
```

How many static objects were created?

Only the first one must create the static flyweights on the heap.

Referenced by library_init(), and ~library_init().

The documentation for this class was generated from the following files:

- ex.h
- utils.cpp

8.89 GiNaC::make_flat_inserter Class Reference

Class to handle the renaming of dummy indices.

```
#include <expairseq.h>
```

Public Member Functions

- make_flat_inserter (const epvector &epv, bool b)
- make_flat_inserter (const exvector &v, bool b)
- ex handle_factor (const ex &x, const ex &coeff)

Private Member Functions

- void combine_indices (const exvector &dummies_of_factor)

Private Attributes

- bool do_renaming
- exvector used_indices

8.89.1 Detailed Description

Class to handle the renaming of dummy indices.

It holds a vector of indices that are being used in the expression so far. If the same index occurs again as a dummy index in a factor, it is to be renamed. Unless dummy index renaming was switched off, of course ;-) .

8.89.2 Constructor & Destructor Documentation

8.89.2.1 make_flat_inserter() [1/2]

```
GiNaC::make_flat_inserter::make_flat_inserter (
            const epvector & epv,
            bool b)  [inline]
```

References GiNaC::are_ex_trivially_equal(), combine_indices(), and do_renaming.

8.89.2.2 make_flat_inserter() [2/2]

```
GiNaC::make_flat_inserter::make_flat_inserter (
            const exvector & v,
            bool b)  [inline]
```

References combine_indices(), and do_renaming.

8.89.3 Member Function Documentation

8.89.3.1 handle_factor()

```
ex GiNaC::make_flat_inserter::handle_factor (
            const ex & x,
            const ex & coeff)  [inline]
```

References GiNaC::coeff(), combine_indices(), do_renaming, GiNaC::get_all_dummy_indices_safely(), GiNaC::ex::get_free_indices(), GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::rename_dummy_indices_uniquely(), and used_indices.

Referenced by GiNaC::ncmul::eval(), GiNaC::expairseq::make_flat(), and GiNaC::expairseq::make_flat().

8.89.3.2 combine_indices()

```
void GiNaC::make_flat_inserter::combine_indices (
            const exvector & dummies_of_factor)  [inline], [private]
```

References used_indices.

Referenced by handle_factor(), make_flat_inserter(), and make_flat_inserter().

8.89.4 Member Data Documentation

8.89.4.1 do_renaming

```
bool GiNaC::make_flat_inserter::do_renaming  [private]
```

Referenced by handle_factor(), make_flat_inserter(), and make_flat_inserter().

8.89.4.2 used_indices

```
exvector GiNaC::make_flat_inserter::used_indices  [private]
```

Referenced by combine_indices(), and handle_factor().

The documentation for this class was generated from the following file:

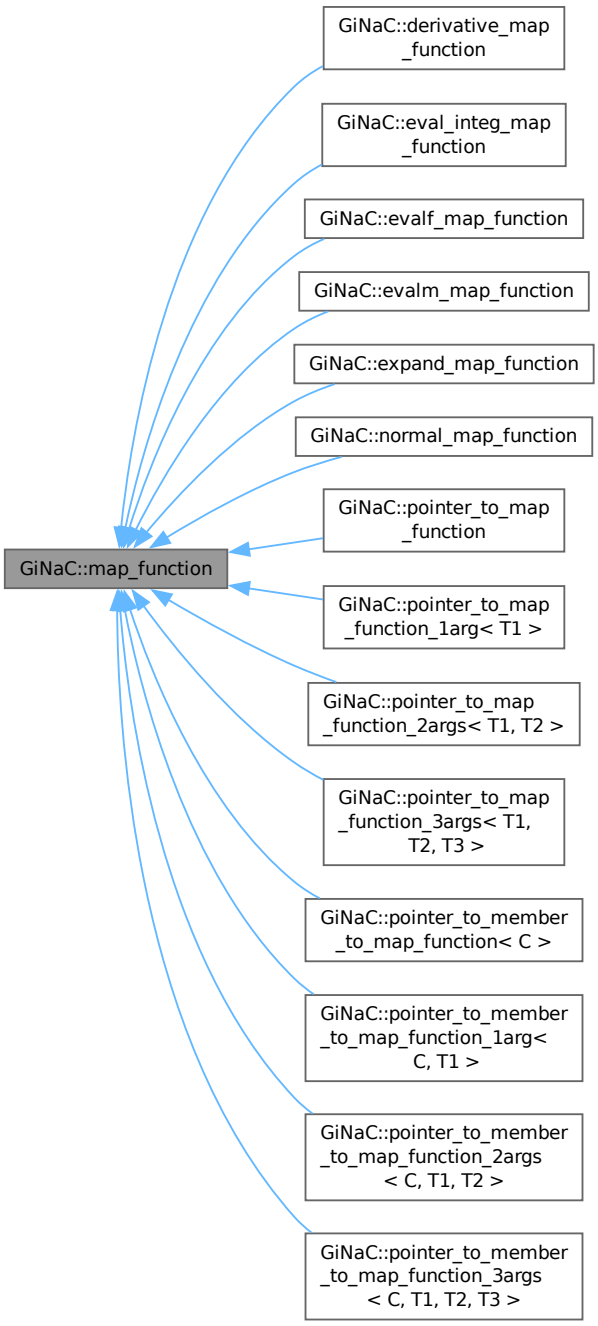
- expairseq.h

8.90 GiNaC::map_function Struct Reference

Function object for map().

```
#include <basic.h>
```

Inheritance diagram for GiNaC::map_function:



Public Types

- typedef const ex & argument_type
- typedef ex result_type

Public Member Functions

- virtual ~map_function ()
- virtual ex operator() (const ex &e)=0

8.90.1 Detailed Description

Function object for map().

8.90.2 Member Typedef Documentation

8.90.2.1 argument_type

```
typedef const ex& GiNaC::map_function::argument_type
```

8.90.2.2 result_type

```
typedef ex GiNaC::map_function::result_type
```

8.90.3 Constructor & Destructor Documentation

8.90.3.1 ~map_function()

```
virtual GiNaC::map_function::~map_function ()  [inline], [virtual]
```

8.90.4 Member Function Documentation

8.90.4.1 operator()()

```
virtual ex GiNaC::map_function::operator() (
            const ex & e)  [pure virtual]
```

Implemented in GiNaC::derivative_map_function, GiNaC::eval_integ_map_function, GiNaC::evalf_map_function, GiNaC::evalm_map_function, GiNaC::expand_map_function, GiNaC::normal_map_function, GiNaC::pointer_to_map_function, GiNaC::pointer_to_map_function_1arg< T1 >, GiNaC::pointer_to_map_function_2args< T1, T2 >, GiNaC::pointer_to_map_functio GiNaC::pointer_to_member_to_map_function< C >, GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >, GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >, and GiNaC::pointer_to_member_to_map_function_3args< C, T

The documentation for this struct was generated from the following file:

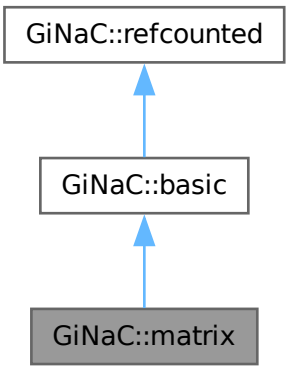
- basic.h

8.91 GiNaC::matrix Class Reference

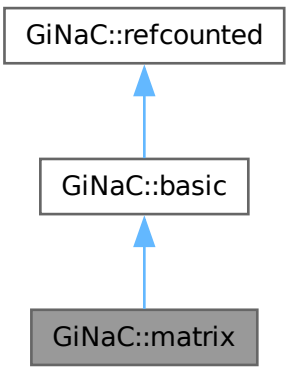
Symbolic matrices.

`#include <matrix.h>`

Inheritance diagram for GiNaC::matrix:



Collaboration diagram for GiNaC::matrix:



Public Member Functions

- matrix (unsigned r, unsigned c)

  Very common ctor.
- matrix (unsigned r, unsigned c, const lst &l)

Construct matrix from (flat) list of elements.

- matrix (std::initializer_list< std::initializer_list< ex > > l)

  Construct a matrix from an 2 dimensional initializer list.
- size_t nops () const override

  nops is defined to be rows x columns.
- ex op (size_t i) const override

  returns matrix entry at position (i/col, icol).
- ex & let_op (size_t i) override

  returns writable matrix entry at position (i/col, icol).
- ex evalm () const override

  Evaluate sums, products and integer powers of matrices.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of an indexed matrix.
- ex add_indexed (const ex &self, const ex &other) const override

  Sum of two indexed matrices.
- ex scalar_mul_indexed (const ex &self, const numeric &other) const override

  Product of an indexed matrix with a number.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of an indexed matrix with something else.
- ex conjugate () const override

  Complex conjugate every matrix entry.
- ex real_part () const override
- ex imag_part () const override
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- unsigned rows () const

  Get number of rows.
- unsigned cols () const

  Get number of columns.
- matrix add (const matrix &other) const

  Sum of matrices.
- matrix sub (const matrix &other) const

  Difference of matrices.
- matrix mul (const matrix &other) const

  Product of matrices.
- matrix mul (const numeric &other) const

  Product of matrix and scalar.
- matrix mul_scalar (const ex &other) const

  Product of matrix and scalar expression.
- matrix pow (const ex &expn) const

  Power of a matrix.
- const ex & operator() (unsigned ro, unsigned co) const

  operator() to access elements for reading.
- ex & operator() (unsigned ro, unsigned co)

  operator() to access elements for writing.
- matrix & set (unsigned ro, unsigned co, const ex &value)
- matrix transpose () const

Transposed of an m x n matrix, producing a new n x m matrix object that represents the transposed.

- ex determinant (unsigned algo=determinant_algo::automatic) const

  Determinant of square matrix.
- ex trace () const

  Trace of a matrix.
- ex charpoly (const ex &lambda) const

  Characteristic Polynomial.
- matrix inverse () const

  Inverse of this matrix, with automatic algorithm selection.
- matrix inverse (unsigned algo) const

  Inverse of this matrix.
- matrix solve (const matrix &vars, const matrix &rhs, unsigned algo=solve_algo::automatic) const

  Solve a linear system consisting of a m x n matrix and a m x p right hand side by applying an elimination scheme to the augmented matrix.
- unsigned rank () const

  Compute the rank of this matrix.
- unsigned rank (unsigned solve_algo) const

  Compute the rank of this matrix using the given algorithm, which should be a member of enum solve_algo.
- bool is_zero_matrix () const

  Function to check that all elements of the matrix are zero.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual return_type_t return_type_tinfo () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- matrix (unsigned r, unsigned c, const exvector &m2)

  Ctor from representation, for internal use only.
- matrix (unsigned r, unsigned c, exvector &&m2)
- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- unsigned return_type () const override
- ex determinant_minor () const

  Recursive determinant for small matrices having at least one symbolic entry.
- std::vector< unsigned > echelon_form (unsigned algo, int n)
- int gauss_elimination (const bool det=false)

  Perform the steps of an ordinary Gaussian elimination to bring the m x n matrix into an upper echelon form.
- int division_free_elimination (const bool det=false)

  Perform the steps of division free elimination to bring the m x n matrix into an upper echelon form.
- int fraction_free_elimination (const bool det=false)

  Perform the steps of Bareiss' one-step fraction free elimination to bring the matrix into an upper echelon form.
- std::vector< unsigned > markowitz_elimination (unsigned n)
- int pivot (unsigned ro, unsigned co, bool symbolic=true)

  Partial pivoting method for matrix elimination schemes.
- void print_elements (const print_context &c, const char *row_start, const char *row_end, const char *row_sep, const char *col_sep) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- unsigned row
  number of rows
- unsigned col
  number of columns
- exvector m
  representation (cols indexed first)

Protected Attributes inherited from GiNaC::basic

- unsigned flags
  of type status_flags
- unsigned hashvalue
  hash value

8.91.1 Detailed Description

Symbolic matrices.

8.91.2 Constructor & Destructor Documentation

8.91.2.1 matrix() [1/5]

```
GiNaC::matrix::matrix (
            unsigned r,
            unsigned c)
```

Very common ctor.

Initializes to r x c-dimensional zero-matrix.

Parameters

r	number of rows
c	number of cols

References GiNaC::_ex0, col, m, matrix(), GiNaC::status_flags::not_shareable, row, and GiNaC::basic::setflag().

Referenced by add(), add_indexed(), charpoly(), GiNaC::clifford_moebius_map(), conjugate(), contract_with(), determinant(), fraction_free_elimination(), imag_part(), inverse(), inverse(), GiNaC::lst_to_clifford(), match_same_type(), matrix(), mul(), mul(), mul_scalar(), pow(), rank(), real_part(), scalar_mul_indexed(), set(), solve(), sub(), subs(), and transpose().

8.91.2.2 matrix() [2/5]

```
GiNaC::matrix::matrix (
            unsigned r,
            unsigned c,
            const lst & l)
```

Construct matrix from (flat) list of elements.

If the list has fewer elements than the matrix, the remaining matrix elements are set to zero. If the list has more elements than the matrix, the excessive elements are thrown away.

References GiNaC::_ex0, col, m, GiNaC::status_flags::not_shareable, row, and GiNaC::basic::setflag().

8.91.2.3 matrix() [3/5]

```
GiNaC::matrix::matrix (
            std::initializer_list< std::initializer_list< ex > > l)
```

Construct a matrix from an 2 dimensional initializer list.

Throws an exception if some row has a different length than all the others.

References col, m, GiNaC::status_flags::not_shareable, row, and GiNaC::basic::setflag().

8.91.2.4 matrix() [4/5]

```
GiNaC::matrix::matrix (
            unsigned r,
            unsigned c,
            const exvector & m2)  [protected]
```

Ctor from representation, for internal use only.

References col, m, GiNaC::status_flags::not_shareable, row, and GiNaC::basic::setflag().

8.91.2.5 matrix() [5/5]

```
GiNaC::matrix::matrix (
            unsigned r,
            unsigned c,
            exvector && m2)  [protected]
```

References col, m, GiNaC::status_flags::not_shareable, row, and GiNaC::basic::setflag().

8.91.3 Member Function Documentation

8.91.3.1 nops()

```
size_t GiNaC::matrix::nops () const  [override], [virtual]
```

nops is defined to be rows x columns.

Reimplemented from GiNaC::basic.

References col, and row.

Referenced by let_op(), GiNaC::nops(), and op().

8.91.3.2 op()

```
ex GiNaC::matrix::op (
            size_t i) const  [override], [virtual]
```

returns matrix entry at position (i/col, icol).

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GINAC_ASSERT, m, and nops().

Referenced by contract_with().

8.91.3.3 let_op()

```
ex & GiNaC::matrix::let_op (
            size_t i)  [override], [virtual]
```

returns writable matrix entry at position (i/col, icol).

Reimplemented from GiNaC::basic.

References GiNaC::basic::ensure_if_modifiable(), GiNaC::basic::ex, GINAC_ASSERT, m, and nops().

8.91.3.4 evalm()

```
ex GiNaC::matrix::evalm () const  [inline], [override], [virtual]
```

Evaluate sums, products and integer powers of matrices.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.91.3.5 subs()

```
ex GiNaC::matrix::subs (
            const exmap & m,
            unsigned options = 0) const  [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References col, GiNaC::basic::ex, m, matrix(), row, and subs().

Referenced by fraction_free_elimination(), and subs().

8.91.3.6 eval_indexed()

```
ex GiNaC::matrix::eval_indexed (
            const basic & i) const  [override], [virtual]
```

Automatic symbolic evaluation of an indexed matrix.

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic(), col, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::idx::get_dim(), GiNaC::idx::get_value(), GINAC_ASSERT, GiNaC::basic::hold(), GiNaC::is_a(), GiNaC::is_dummy_pair(), GiNaC::ex::is_equal(), GiNaC::info_flags::nonnegint, GiNaC::basic::nops(), GiNaC::basic::op(), row, and trace().

8.91.3.7 add_indexed()

```
ex GiNaC::matrix::add_indexed (
            const ex & self,
            const ex & other) const  [override], [virtual]
```

Sum of two indexed matrices.

Reimplemented from GiNaC::basic.

References add(), col, GiNaC::basic::ex, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::indexed::indexed(), GiNaC::is_a(), GiNaC::ex::is_equal(), matrix(), GiNaC::ex::nops(), GiNaC::ex::op(), row, and transpose().

8.91.3.8 scalar_mul_indexed()

```
ex GiNaC::matrix::scalar_mul_indexed (
            const ex & self,
            const numeric & other) const  [override], [virtual]
```

Product of an indexed matrix with a number.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::indexed::indexed(), GiNaC::is_a(), matrix(), mul(), GiNaC::ex::nops(), and GiNaC::ex::op().

8.91.3.9 contract_with()

```
bool GiNaC::matrix::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [override], [virtual]
```

Contraction of an indexed matrix with something else.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, col, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::indexed::indexed(), GiNaC::is_a(), GiNaC::is_dummy_pair(), matrix(), mul(), GiNaC::ex::op(), op(), row, and transpose().

8.91.3.10 conjugate()

```
ex GiNaC::matrix::conjugate () const  [override], [virtual]
```

Complex conjugate every matrix entry.

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), col, GiNaC::basic::ex, m, matrix(), and row.

8.91.3.11 real_part()

```
ex GiNaC::matrix::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References col, GiNaC::basic::ex, m, matrix(), and row.

8.91.3.12 imag_part()

```
ex GiNaC::matrix::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References col, GiNaC::basic::ex, m, matrix(), and row.

8.91.3.13 archive()

```
void GiNaC::matrix::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_ex(), GiNaC::archive_node::add_unsigned(), col, m, and row.

8.91.3.14 read_archive()

```
void GiNaC::matrix::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References col, GiNaC::basic::ex, GiNaC::archive_node::find_ex_by_loc(), GiNaC::archive_node::find_property_range(), GiNaC::archive_node::find_unsigned(), m, and row.

8.91.3.15 match_same_type()

```
bool GiNaC::matrix::match_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic(), col, cols(), GINAC_ASSERT, GiNaC::is_exactly_a(), matrix(), row, and rows().

8.91.3.16 return_type()

```
unsigned GiNaC::matrix::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::noncommutative.

8.91.3.17 rows()

```
unsigned GiNaC::matrix::rows () const  [inline]
```

Get number of rows.

References row.

Referenced by GiNaC::clifford_unit(), division_free_elimination(), fraction_free_elimination(), gauss_elimination(), GiNaC::lsolve(), match_same_type(), mul(), GiNaC::reduced_matrix(), GiNaC::rows(), solve(), GiNaC::sub_matrix(), and transpose().

8.91.3.18 cols()

```
unsigned GiNaC::matrix::cols () const  [inline]
```

Get number of columns.

References col.

Referenced by GiNaC::clifford_unit(), GiNaC::cols(), determinant_minor(), division_free_elimination(), fraction_free_elimination(), gauss_elimination(), GiNaC::lsolve(), match_same_type(), mul(), GiNaC::reduced_matrix(), solve(), GiNaC::sub_matrix(), and transpose().

8.91.3.19 add()

```
matrix GiNaC::matrix::add (
            const matrix & other) const
```

Sum of matrices.

Exceptions

logic_error	(incompatible matrices)

References col, m, matrix(), and row.

Referenced by add_indexed(), and GiNaC::add::evalm().

8.91.3.20 sub()

```
matrix GiNaC::matrix::sub (
            const matrix & other) const
```

Difference of matrices.

Exceptions

logic_error	(incompatible matrices)

References col, m, matrix(), and row.

8.91.3.21 mul() [1/2]

```
matrix GiNaC::matrix::mul (
            const matrix & other) const
```

Product of matrices.

Exceptions

logic_error	(incompatible matrices)

References col, cols(), GiNaC::is_zero(), m, matrix(), row, and rows().

Referenced by contract_with(), GiNaC::ncmul::evalm(), GiNaC::Gt::findMoebiusTransform(), pow(), and scalar_mul_indexed().

8.91.3.22 mul() [2/2]

```
matrix GiNaC::matrix::mul (
            const numeric & other) const
```

Product of matrix and scalar.

References col, m, matrix(), and row.

8.91.3.23 mul_scalar()

```
matrix GiNaC::matrix::mul_scalar (
            const ex & other) const
```

Product of matrix and scalar expression.

References col, GiNaC::return_types::commutative, GiNaC::basic::ex, m, matrix(), GiNaC::ex::return_type(), and row.

Referenced by GiNaC::mul::evalm().

8.91.3.24 pow()

```
matrix GiNaC::matrix::pow (
            const ex & expn) const
```

Power of a matrix.

Currently handles integer exponents only.

References GiNaC::_ex1, GiNaC::_num1_p, GiNaC::_num2_p, col, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::ex::info(), GiNaC::info_flags::integer, inverse(), GiNaC::is_exactly_a(), GiNaC::numeric::is_odd(), GiNaC::numeric::is_zero(), matrix(), mul(), GiNaC::info_flags::negative, and row.

8.91.3.25 operator()() [1/2]

```
const ex & GiNaC::matrix::operator() (
            unsigned ro,
            unsigned co) const
```

operator() to access elements for reading.

Parameters

ro	row of element
co	column of element

Exceptions

range_error	(index out of range)

References col, GiNaC::basic::ex, m, and row.

8.91.3.26 operator()() [2/2]

```
ex & GiNaC::matrix::operator() (
            unsigned ro,
            unsigned co)
```

operator() to access elements for writing.

Parameters

ro	row of element
co	column of element

Exceptions

range_error	(index out of range)

References col, GiNaC::basic::ensure_if_modifiable(), GiNaC::basic::ex, m, and row.

8.91.3.27 set()

```
matrix & GiNaC::matrix::set (
            unsigned ro,
            unsigned co,
            const ex & value)  [inline]
```

References GiNaC::basic::ex, and matrix().

8.91.3.28 transpose()

```
matrix GiNaC::matrix::transpose () const
```

Transposed of an m x n matrix, producing a new n x m matrix object that represents the transposed.

References cols(), m, matrix(), and rows().

Referenced by add_indexed(), contract_with(), and GiNaC::transpose().

8.91.3.29 determinant()

```
ex GiNaC::matrix::determinant (
             unsigned algo = determinant_algo::automatic) const
```

Determinant of square matrix.

This routine doesn't actually calculate the determinant, it only implements some heuristics about which algorithm to run. If all the elements of the matrix are elements of an integral domain the determinant is also in that integral domain and the result is expanded only. If one or more elements are from a quotient field the determinant is usually also in that quotient field and the result is normalized before it is returned. This implies that the determinant of the symbolic 2x2 matrix [[a/(a-b),1],[b/(a-b),1]] is returned as unity. (In this respect, it behaves like MapleV and unlike Mathematica.)

Parameters

algo	allows to chose an algorithm

Returns

the determinant as a new expression

Exceptions

logic_error	(matrix not square)

See also

determinant_algo

References GiNaC::_ex0, GiNaC::determinant_algo::automatic, GiNaC::determinant_algo::bareiss, col, GiNaC::info_flags::crational_p determinant_minor(), GiNaC::determinant_algo::divfree, division_free_elimination(), GiNaC::basic::ex, fraction_free_elimination(), GiNaC::determinant_algo::gauss, gauss_elimination(), GINAC_ASSERT, GiNaC::ex::info(), GiNaC::ex::is_zero(), GiNaC::is_zero(), GiNaC::determinant_algo::laplace, m, matrix(), GiNaC::ex::normal(), GiNaC::info_flags::numeric, GiNaC::permutation_sign(), GiNaC::info_flags::rational_function, and row.

Referenced by charpoly(), GiNaC::tensepsilon::contract_with(), GiNaC::determinant(), and GiNaC::resultant().

8.91.3.30 trace()

```
ex GiNaC::matrix::trace () const
```

Trace of a matrix.

The result is normalized if it is in some quotient field and expanded only otherwise. This implies that the trace of the symbolic 2x2 matrix [[a/(a-b),x],[y,b/(b-a)]] is recognized to be unity.

Returns

the sum of diagonal elements

Exceptions

logic_error	(matrix not square)

References col, GiNaC::info_flags::crational_polynomial, GiNaC::basic::ex, GiNaC::ex::expand(), GiNaC::ex::info(), m, GiNaC::ex::normal(), GiNaC::info_flags::rational_function, and row.

Referenced by charpoly(), eval_indexed(), and GiNaC::trace().

8.91.3.31 charpoly()

```
ex GiNaC::matrix::charpoly (
             const ex & lambda) const
```

Characteristic Polynomial.

Following mathematica notation the characteristic polynomial of a matrix M is defined as the determinant of (M - lambda $*$ 1) where 1 stands for the unit matrix of the same dimension as M. Note that some CASs define it with a sign inside the determinant which gives rise to an overall sign if the dimension is odd. This method returns the characteristic polynomial collected in powers of lambda as a new expression.

Returns

characteristic polynomial as new expression

Exceptions

logic_error	(matrix not square)

See also

matrix::determinant()

References col, GiNaC::ex::collect(), determinant(), GiNaC::basic::ex, GiNaC::ex::ex(), m, matrix(), GiNaC::mul::mul(), GiNaC::info_flags::numeric, GiNaC::power::power(), row, and trace().

Referenced by GiNaC::charpoly().

8.91.3.32 inverse() [1/2]

```
matrix GiNaC::matrix::inverse () const
```

Inverse of this matrix, with automatic algorithm selection.

References GiNaC::solve_algo::automatic, inverse(), and matrix().

Referenced by GiNaC::inverse(), GiNaC::inverse(), inverse(), and pow().

8.91.3.33 inverse() [2/2]

```
matrix GiNaC::matrix::inverse (
            unsigned algo) const
```

Inverse of this matrix.

Parameters

algo	selects the algorithm (one of solve_algo)

Returns

the inverted matrix

Exceptions

logic_error	(matrix not square)
runtime_error	(singular matrix)

References GiNaC::_ex1, col, matrix(), row, solve(), and GiNaC::symbol::symbol().

8.91.3.34 solve()

```
matrix GiNaC::matrix::solve (
            const matrix & vars,
            const matrix & rhs,
            unsigned algo = solve_algo::automatic) const
```

Solve a linear system consisting of a m x n matrix and a m x p right hand side by applying an elimination scheme to the augmented matrix.

Parameters

vars	n x p matrix, all elements must be symbols
rhs	m x p matrix
algo	selects the solving algorithm

Returns

n x p solution matrix

Exceptions

logic_error	(incompatible matrices)
invalid_argument	(1st argument must be matrix of symbols)
runtime_error	(inconsistent linear system)

See also

solve_algo

References cols(), echelon_form(), GiNaC::basic::ex, GiNaC::basic::info(), m, matrix(), GiNaC::basic::normal(), GiNaC::rhs(), rows(), and GiNaC::info_flags::symbol.

Referenced by inverse(), GiNaC::lsolve(), and GiNaC::sqrfree_parfrac().

8.91.3.35 rank() [1/2]

```
unsigned GiNaC::matrix::rank () const
```

Compute the rank of this matrix.

References GiNaC::solve_algo::automatic, and rank().

Referenced by rank(), GiNaC::rank(), and GiNaC::rank().

8.91.3.36 rank() [2/2]

```
unsigned GiNaC::matrix::rank (
            unsigned solve_algo) const
```

Compute the rank of this matrix using the given algorithm, which should be a member of enum solve_algo.

References col, echelon_form(), GINAC_ASSERT, m, matrix(), and row.

8.91.3.37 is_zero_matrix()

```
bool GiNaC::matrix::is_zero_matrix () const
```

Function to check that all elements of the matrix are zero.

References m.

8.91.3.38 determinant_minor()

```
ex GiNaC::matrix::determinant_minor () const  [protected]
```

Recursive determinant for small matrices having at least one symbolic entry.

The basic algorithm, known as Laplace-expansion, is enhanced by some bookkeeping to avoid calculation of the same submatrices ("minors") more than once. According to W.M.Gentleman and S.C.Johnson this algorithm is better than elimination schemes for matrices of sparse multivariate polynomials and also for matrices of dense univariate polynomials if the matrix' dimension is larger than 7.

Returns

the determinant as a new expression (in expanded form)

See also

matrix::determinant()

References GiNaC::_ex0, GiNaC::_ex1, cols(), GiNaC::basic::ex, GiNaC::ex::expand(), GiNaC::ex::is_zero(), GiNaC::is_zero(), and m.

Referenced by determinant().

8.91.3.39 echelon_form()

```
std::vector< unsigned > GiNaC::matrix::echelon_form (
            unsigned algo,
            int n)  [protected]
```

References GiNaC::solve_algo::automatic, GiNaC::solve_algo::bareiss, col, GiNaC::solve_algo::divfree, division_free_elimination(), fraction_free_elimination(), GiNaC::solve_algo::gauss, gauss_elimination(), m, GiNaC::solve_algo::markowitz, markowitz_elimination(), GiNaC::info_flags::numeric, and row.

Referenced by rank(), and solve().

8.91.3.40 gauss_elimination()

```
int GiNaC::matrix::gauss_elimination (
            const bool det = false)  [protected]
```

Perform the steps of an ordinary Gaussian elimination to bring the m x n matrix into an upper echelon form.

The algorithm is ok for matrices with numeric coefficients but quite unsuited for symbolic matrices.

Parameters

det	may be set to true to save a lot of space if one is only interested in the diagonal elements (i.e. for calculating determinants). The others are set to zero in this case.

Returns

sign is 1 if an even number of rows was swapped, -1 if an odd number of rows was swapped and 0 if the matrix is singular.

References GiNaC::_ex0, cols(), GiNaC::basic::ensure_if_modifiable(), GiNaC::basic::ex, GINAC_ASSERT, GiNaC::is_zero(), m, GiNaC::basic::normal(), pivot(), and rows().

Referenced by determinant(), and echelon_form().

8.91.3.41 division_free_elimination()

```
int GiNaC::matrix::division_free_elimination (
            const bool det = false)  [protected]
```

Perform the steps of division free elimination to bring the m x n matrix into an upper echelon form.

Parameters

det	may be set to true to save a lot of space if one is only interested in the diagonal elements (i.e. for calculating determinants). The others are set to zero in this case.

Returns

sign is 1 if an even number of rows was swapped, -1 if an odd number of rows was swapped and 0 if the matrix is singular.

References GiNaC::_ex0, cols(), GiNaC::basic::ensure_if_modifiable(), GINAC_ASSERT, m, pivot(), and rows().

Referenced by determinant(), and echelon_form().

8.91.3.42 fraction_free_elimination()

```
int GiNaC::matrix::fraction_free_elimination (
            const bool det = false)  [protected]
```

Perform the steps of Bareiss' one-step fraction free elimination to bring the matrix into an upper echelon form.

Fraction free elimination means that divide is used straightforwardly, without computing GCDs first. This is possible, since we know the divisor at each step.

Parameters

det	may be set to true to save a lot of space if one is only interested in the last element (i.e. for calculating determinants). The others are set to zero in this case.

Returns

sign is 1 if an even number of rows was swapped, -1 if an odd number of rows was swapped and 0 if the matrix is singular.

References GiNaC::_ex0, GiNaC::_ex1, cols(), GiNaC::divide(), GiNaC::basic::ensure_if_modifiable(), GiNaC::basic::ex, GiNaC::basic::expand(), GINAC_ASSERT, GiNaC::is_zero(), m, matrix(), GiNaC::subs_options::no_pattern, GiNaC::ex::op(), rows(), and subs().

Referenced by determinant(), and echelon_form().

8.91.3.43 markowitz_elimination()

```
std::vector< unsigned > GiNaC::matrix::markowitz_elimination (
            unsigned n)  [protected]
```

References GiNaC::_ex0, col, GiNaC::basic::ex, GINAC_ASSERT, GiNaC::ex::is_zero(), GiNaC::is_zero(), m, GiNaC::basic::normal(), row, and std::swap().

Referenced by echelon_form().

8.91.3.44 pivot()

```
int GiNaC::matrix::pivot (
            unsigned ro,
            unsigned co,
            bool symbolic = true)  [protected]
```

Partial pivoting method for matrix elimination schemes.

Usual pivoting (symbolic==false) returns the index to the element with the largest absolute value in column ro and swaps the current row with the one where the element was found. With (symbolic==true) it does the same thing with the first non-zero element.

Parameters

ro	is the row from where to begin
co	is the column to be inspected
symbolic	signal if we want the first non-zero element to be pivoted (true) or the one with the largest absolute value (false).

Returns

0 if no interchange occurred, -1 if all are zero (usually signaling a degeneracy) and positive integer k means that rows ro and k were swapped.

References GiNaC::abs(), col, GiNaC::basic::ensure_if_modifiable(), GiNaC::ex_to(), GiNaC::basic::expand(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::is_zero(), GiNaC::numeric::is_zero(), m, row, and GiNaC::swap().

Referenced by division_free_elimination(), and gauss_elimination().

8.91.3.45 print_elements()

```
void GiNaC::matrix::print_elements (
            const print_context & c,
            const char * row_start,
            const char * row_end,
            const char * row_sep,
            const char * col_sep) const  [protected]
```

References col, m, row, and GiNaC::print_context::s.

Referenced by do_print(), do_print_latex(), and do_print_python_repr().

8.91.3.46 do_print()

```
void GiNaC::matrix::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References print_elements(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.91.3.47 do_print_latex()

```
void GiNaC::matrix::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

References col, print_elements(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.91.3.48 do_print_python_repr()

```
void GiNaC::matrix::do_print_python_repr (
            const print_python_repr & c,
            unsigned level) const  [protected]
```

References print_elements(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.91.4 Member Data Documentation

8.91.4.1 row

```
unsigned GiNaC::matrix::row  [protected]
```

number of rows

Referenced by add(), add_indexed(), archive(), charpoly(), conjugate(), contract_with(), determinant(), echelon_form(), eval_indexed(), imag_part(), inverse(), markowitz_elimination(), match_same_type(), matrix(), matrix(), matrix(), matrix(), matrix(), mul(), mul(), mul_scalar(), nops(), operator()(), operator()(), pivot(), pow(), print_elements(), rank(), read_archive(), real_part(), rows(), sub(), subs(), and trace().

8.91.4.2 col

```
unsigned GiNaC::matrix::col  [protected]
```

number of columns

Referenced by add(), add_indexed(), archive(), charpoly(), cols(), conjugate(), contract_with(), determinant(), do_print_latex(), echelon_form(), eval_indexed(), imag_part(), inverse(), markowitz_elimination(), match_same_type(), matrix(), matrix(), matrix(), matrix(), matrix(), mul(), mul(), mul_scalar(), nops(), operator()(), operator()(), pivot(), pow(), print_elements(), rank(), read_archive(), real_part(), sub(), subs(), and trace().

8.91.4.3 m

```
exvector GiNaC::matrix::m  [protected]
```

representation (cols indexed first)

Referenced by add(), archive(), charpoly(), conjugate(), determinant(), determinant_minor(), division_free_elimination(), echelon_form(), fraction_free_elimination(), gauss_elimination(), imag_part(), is_zero_matrix(), let_op(), markowitz_elimination(), matrix(), matrix(), matrix(), matrix(), matrix(), mul(), mul(), mul_scalar(), op(), operator()(), operator()(), pivot(), print_elements(), rank(), read_archive(), real_part(), solve(), sub(), subs(), trace(), and transpose().

The documentation for this class was generated from the following files:

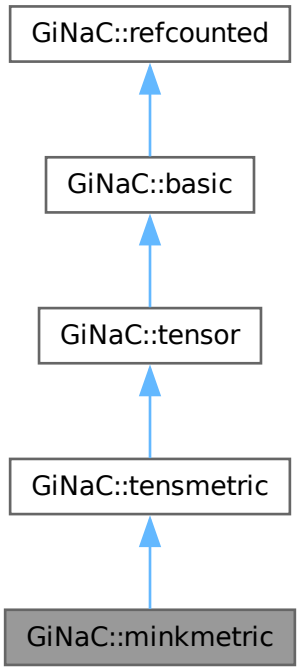
- matrix.h
- matrix.cpp

8.92 GiNaC::minkmetric Class Reference

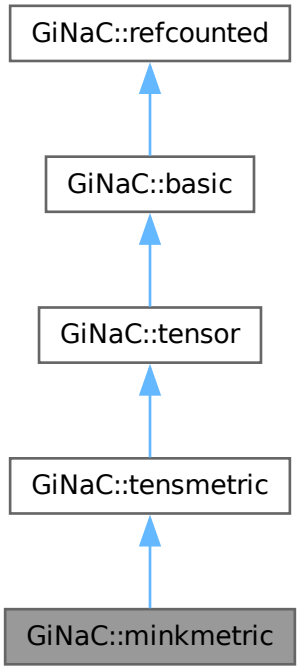
This class represents a Minkowski metric tensor.

```
#include <tensor.h>
```

Inheritance diagram for GiNaC::minkmetric:



Collaboration diagram for GiNaC::minkmetric:



Public Member Functions

- minkmetric (bool pos_sig)

  Construct Lorentz metric tensor with given signature.
- bool info (unsigned inf) const override

  Information about the object.
- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of an indexed Lorentz metric tensor.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.

Public Member Functions inherited from GiNaC::tensmetric

- bool info (unsigned inf) const override

  Information about the object.
- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of an indexed metric tensor.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of an indexed metric tensor with something else.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

Check whether this is a polynomial in the given variables.

- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- unsigned return_type () const override
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensmetric

- unsigned return_type () const override
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Attributes

- bool pos_sig

  If true, the metric is diag(-1,1,1...).

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags
  of type status_flags
- unsigned hashvalue
  hash value

8.92.1 Detailed Description

This class represents a Minkowski metric tensor.

It has all the properties of a metric tensor and is (as a matrix) equal to diag(1,-1,-1,...) or diag(-1,1,1,...).

8.92.2 Constructor & Destructor Documentation

8.92.2.1 minkmetric()

```
GiNaC::minkmetric::minkmetric (
            bool pos_sig)
```

Construct Lorentz metric tensor with given signature.

References minkmetric(), and pos_sig.

Referenced by minkmetric().

8.92.3 Member Function Documentation

8.92.3.1 info()

```
bool GiNaC::minkmetric::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References GiNaC::info_flags::real.

8.92.3.2 eval_indexed()

```
ex GiNaC::minkmetric::eval_indexed (
            const basic & i) const  [override], [virtual]
```

Automatic symbolic evaluation of an indexed Lorentz metric tensor.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::_ex_1, GiNaC::ex_to(), GiNaC::idx::get_value(), GINAC_ASSERT, GiNaC::is_a(), GiNaC::info_flags::nonnegint, GiNaC::basic::nops(), GiNaC::basic::op(), and pos_sig.

8.92.3.3 archive()

```
void GiNaC::minkmetric::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_bool(), and pos_sig.

8.92.3.4 read_archive()

```
void GiNaC::minkmetric::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::find_bool(), and pos_sig.

8.92.3.5 return_type()

```
unsigned GiNaC::minkmetric::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative.

8.92.3.6 do_print()

```
void GiNaC::minkmetric::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

8.92.3.7 do_print_latex()

```
void GiNaC::minkmetric::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

8.92.4 Member Data Documentation

8.92.4.1 pos_sig

```
bool GiNaC::minkmetric::pos_sig  [private]
```

If true, the metric is diag(-1,1,1...).

Otherwise it is diag(1,-1,-1,...).

Referenced by archive(), eval_indexed(), minkmetric(), and read_archive().

The documentation for this class was generated from the following files:

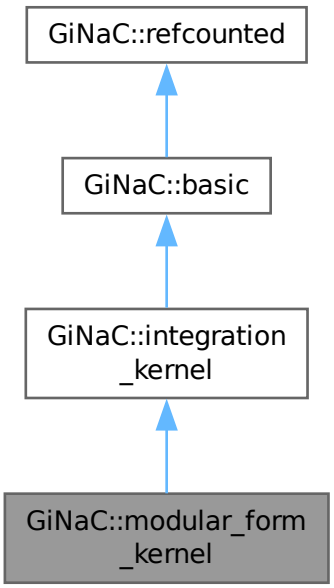
- tensor.h
- tensor.cpp

8.93 GiNaC::modular_form_kernel Class Reference

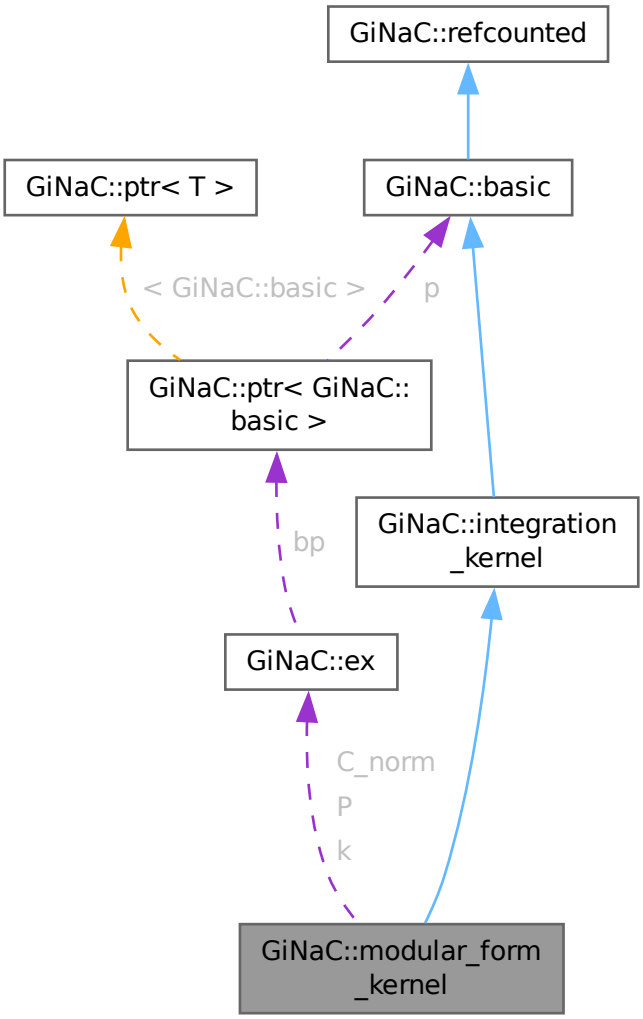
A kernel corresponding to a polynomial in Eisenstein series.

```
#include <integration_kernel.h>
```

Inheritance diagram for GiNaC::modular_form_kernel:



Collaboration diagram for GiNaC::modular_form_kernel:



Public Member Functions

- modular_form_kernel (const ex &k, const ex &P, const ex &C_norm=numeric(1))
- ex series (const relational &r, int order, unsigned options=0) const override

  The series method for this class returns the qbar-expansion of the modular form, without an additional factor of C_norm/qbar.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- bool is_numeric (void) const override

  This routine returns true, if the integration kernel can be evaluated numerically.

- ex Laurent_series (const ex &qbar, int order) const override

  Returns the Laurent series, starting possibly with the pole term.
- ex get_numerical_value (const ex &qbar, int N_trunc=0) const override

  Returns the value of the modular form.
- ex q_expansion_modular_form (const ex &q, int order) const

Public Member Functions inherited from GiNaC::integration_kernel

- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const

- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

Like print(), but dispatch to the specified class.

- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- bool uses_Laurent_series () const override

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual cln::cl_N series_coeff_impl (int i) const

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex k
- ex P
- ex C_norm

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.93.1 Detailed Description

A kernel corresponding to a polynomial in Eisenstein series.

This class represents the differential one-form

$$\omega^{\mathrm{modular}}(P_k(\eta_{k_1}^{(1)},\ldots,\eta_{k_r}^{(r)})) = C_k P_k(\eta_{k_1}^{(1)},\ldots,\eta_{k_r}^{(r)})\frac{d\bar{q}_N}{\bar{q}_N}.$$

8.93.2 Constructor & Destructor Documentation

8.93.2.1 modular_form_kernel()

```
GiNaC::modular_form_kernel::modular_form_kernel (
             const ex & k,
             const ex & P,
             const ex & C_norm = numeric(1))
```

References C_norm, k, modular_form_kernel(), and P.

Referenced by modular_form_kernel().

8.93.3 Member Function Documentation

8.93.3.1 series()

```
ex GiNaC::modular_form_kernel::series (
             const relational & r,
             int order,
             unsigned options = 0) const  [override], [virtual]
```

The series method for this class returns the qbar-expansion of the modular form, without an additional factor of C_norm/qbar.

Reimplemented from GiNaC::integration_kernel.

References GiNaC::relational::lhs(), P, GiNaC::pow(), GiNaC::relational::rhs(), and GiNaC::ex::series().

Referenced by q_expansion_modular_form().

8.93.3.2 nops()

```
size_t GiNaC::modular_form_kernel::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.93.3.3 op()

```
ex GiNaC::modular_form_kernel::op (
             size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References C_norm, k, and P.

8.93.3.4 let_op()

```
ex & GiNaC::modular_form_kernel::let_op (
             size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References C_norm, GiNaC::basic::ensure_if_modifiable(), k, and P.

8.93.3.5 is_numeric()

```
bool GiNaC::modular_form_kernel::is_numeric (
             void ) const  [override], [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented from GiNaC::integration_kernel.

References C_norm, GiNaC::ex::evalf(), GiNaC::ex::info(), k, GiNaC::info_flags::nonnegint, GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), q_expansion_modular_form(), GiNaC::series_to_poly(), and GiNaC::ex::subs().

8.93.3.6 Laurent_series()

```
ex GiNaC::modular_form_kernel::Laurent_series (
             const ex & x,
             int order) const  [override], [virtual]
```

Returns the Laurent series, starting possibly with the pole term.

Neglected terms are of order $O(x^order)$.

Reimplemented from GiNaC::integration_kernel.

References C_norm, q_expansion_modular_form(), GiNaC::ex::series(), and GiNaC::series_to_poly().

8.93.3.7 get_numerical_value()

```
ex GiNaC::modular_form_kernel::get_numerical_value (
             const ex & qbar,
             int N_trunc = 0) const  [override], [virtual]
```

Returns the value of the modular form.

Reimplemented from GiNaC::integration_kernel.

References C_norm, GiNaC::integration_kernel::get_numerical_value_impl(), and GiNaC::numeric::numeric().

8.93.3.8 uses_Laurent_series()

```
bool GiNaC::modular_form_kernel::uses_Laurent_series () const  [override], [protected], [virtual]
```

Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).

Returns false if this is not the case (and the class has an implementation of series_coeff_impl).

Reimplemented from GiNaC::integration_kernel.

8.93.3.9 q_expansion_modular_form()

```
ex GiNaC::modular_form_kernel::q_expansion_modular_form (
            const ex & q,
            int order) const
```

References series().

Referenced by is_numeric(), and Laurent_series().

8.93.3.10 do_print()

```
void GiNaC::modular_form_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References C_norm, k, P, GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.93.4 Member Data Documentation

8.93.4.1 k

```
ex GiNaC::modular_form_kernel::k  [protected]
```

Referenced by do_print(), is_numeric(), let_op(), modular_form_kernel(), and op().

8.93.4.2 P

```
ex GiNaC::modular_form_kernel::P  [protected]
```

Referenced by do_print(), let_op(), modular_form_kernel(), op(), and series().

8.93.4.3 C_norm

```
ex GiNaC::modular_form_kernel::C_norm  [protected]
```

Referenced by do_print(), get_numerical_value(), is_numeric(), Laurent_series(), let_op(), modular_form_kernel(), and op().

The documentation for this class was generated from the following files:

- integration_kernel.h
- integration_kernel.cpp

8.94 GiNaC::basic_partition_generator::mpartition2 Struct Reference

```
#include <utils.h>
```

Public Member Functions

- mpartition2 (unsigned n_, unsigned m_)
- bool next_partition ()

Public Attributes

- std::vector< unsigned > x
- unsigned n
- unsigned m

8.94.1 Constructor & Destructor Documentation

8.94.1.1 mpartition2()

```
GiNaC::basic_partition_generator::mpartition2::mpartition2 (
            unsigned n_,
            unsigned m_)  [inline]
```

References m, n, and x.

8.94.2 Member Function Documentation

8.94.2.1 next_partition()

```
bool GiNaC::basic_partition_generator::mpartition2::next_partition ()  [inline]
```

References m, and x.

Referenced by GiNaC::partition_generator::next(), and GiNaC::partition_with_zero_parts_generator::next().

8.94.3 Member Data Documentation

8.94.3.1 x

```
std::vector<unsigned> GiNaC::basic_partition_generator::mpartition2::x
```

Referenced by GiNaC::partition_generator::get(), GiNaC::partition_with_zero_parts_generator::get(), mpartition2(), and next_partition().

8.94.3.2 n

```
unsigned GiNaC::basic_partition_generator::mpartition2::n
```

Referenced by mpartition2(), and GiNaC::partition_with_zero_parts_generator::next().

8.94.3.3 m

```
unsigned GiNaC::basic_partition_generator::mpartition2::m
```

Referenced by GiNaC::partition_generator::get(), GiNaC::partition_with_zero_parts_generator::get(), mpartition2(), GiNaC::partition_with_zero_parts_generator::next(), and next_partition().

The documentation for this struct was generated from the following file:

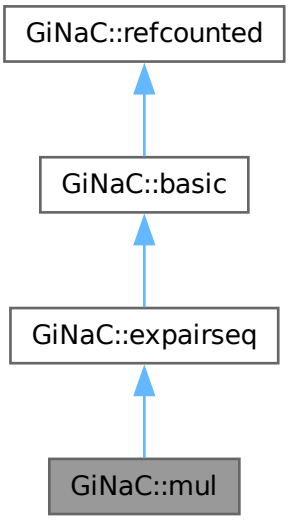
- utils.h

8.95 GiNaC::mul Class Reference

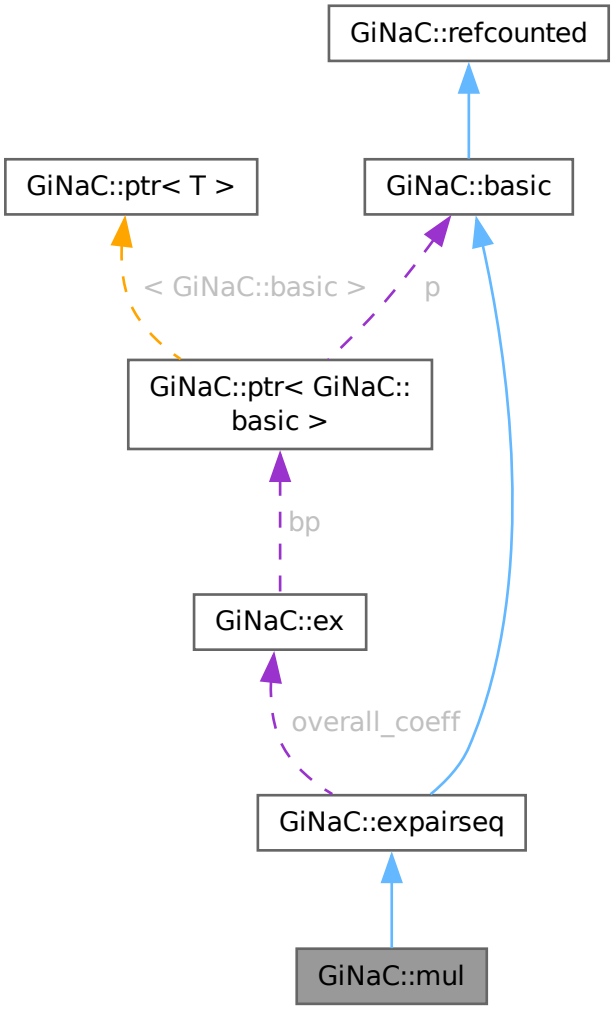
Product of expressions.

```
#include <mul.h>
```

Inheritance diagram for GiNaC::mul:



Collaboration diagram for GiNaC::mul:



Public Member Functions

- mul (const ex &lh, const ex &rh)
- mul (const exvector &v)
- mul (const epvector &v)
- mul (const epvector &v, const ex &oc, bool do_index_renaming=false)
- mul (epvector &&vp)
- mul (epvector &&vp, const ex &oc, bool do_index_renaming=false)
- mul (const ex &lh, const ex &mh, const ex &rh)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- bool is_polynomial (const ex &var) const override

Check whether this is a polynomial in the given variables.

- int degree (const ex &s) const override

  Return degree of highest power in object s.
- int ldegree (const ex &s) const override

  Return degree of lowest power in object s.
- ex coeff (const ex &s, int n=1) const override

  Return coefficient of degree n in object s.
- bool has (const ex &other, unsigned options=0) const override

  Test for occurrence of a pattern.
- ex eval () const override

  Perform automatic term rewriting rules in this class.
- ex evalf () const override

  Evaluate object numerically.
- ex real_part () const override
- ex imag_part () const override
- ex evalm () const override

  Evaluate sums, products and integer powers of matrices.
- ex series (const relational &s, int order, unsigned options=0) const override

  Implementation of ex::series() for product.
- ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const override

  Implementation of ex::normal() for a product.
- numeric integer_content () const override
- ex smod (const numeric &xi) const override

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- numeric max_coefficient () const override

  Implementation ex::max_coefficient().
- exvector get_free_indices () const override

  Return a vector containing the free indices of an expression.
- ex conjugate () const override
- ex algebraic_subs_mul (const exmap &m, unsigned options) const

Public Member Functions inherited from GiNaC::expairseq

- expairseq (const ex &lh, const ex &rh)
- expairseq (const exvector &v)
- expairseq (const epvector &v, const ex &oc, bool do_index_renaming=false)
- expairseq (epvector &&vp, const ex &oc, bool do_index_renaming=false)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex map (map_function &f) const override

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- ex to_rational (exmap &repl) const override

  Implementation of ex::to_rational() for expairseqs.
- ex to_polynomial (exmap &repl) const override

  Implementation of ex::to_polynomial() for expairseqs.
- bool match (const ex &pattern, exmap &repl_lst) const override

  Check whether the expression matches a given pattern.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- void archive (archive_node &n) const override

  Save (serialize) the object into archive node.
- void read_archive (const archive_node &n, lst &syms) override

  Load (deserialize) the object from an archive node.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual void accept (GiNaC::visitor &v) const
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for a product.
- ex eval_ncmul (const exvector &v) const override
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- ex thisexpairseq (const epvector &v, const ex &oc, bool do_index_renaming=false) const override

  Create an object of this type.
- ex thisexpairseq (epvector &&vp, const ex &oc, bool do_index_renaming=false) const override
- expair split_ex_to_pair (const ex &e) const override

  Form an expair from an ex, using the corresponding semantics.
- expair combine_ex_with_coeff_to_pair (const ex &e, const ex &c) const override
- expair combine_pair_with_coeff_to_pair (const expair &p, const ex &c) const override
- ex recombine_pair_to_ex (const expair &p) const override

  Form an ex out of an expair, using the corresponding semantics.
- bool expair_needs_further_processing (epp it) override
- ex default_overall_coeff () const override
- void combine_overall_coeff (const ex &c) override
- void combine_overall_coeff (const ex &c1, const ex &c2) override
- bool can_make_flat (const expair &p) const override
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- void find_real_imag (ex &, ex &) const
- void print_overall_coeff (const print_context &c, const char *mul_sym) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_csrc (const print_csrc &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const
- epvector expandchildren (unsigned options) const

  Member-wise expand the expairs representing this sequence.

Protected Member Functions inherited from GiNaC::expairseq

- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- virtual void printseq (const print_context &c, char delim, unsigned this_precedence, unsigned upper_precedence) const
- virtual void printpair (const print_context &c, const expair &p, unsigned upper_precedence) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void construct_from_2_ex (const ex &lh, const ex &rh)

- void construct_from_2_expairseq (const expairseq &s1, const expairseq &s2)
- void construct_from_expairseq_ex (const expairseq &s, const ex &e)
- void construct_from_exvector (const exvector &v)
- void construct_from_epvector (const epvector &v, bool do_index_renaming=false)
- void construct_from_epvector (epvector &&v, bool do_index_renaming=false)
- void make_flat (const exvector &v)

  Combine this expairseq with argument exvector.
- void make_flat (const epvector &v, bool do_index_renaming=false)

  Combine this expairseq with argument epvector.
- void canonicalize ()

  Brings this expairseq into a sorted (canonical) form.
- void combine_same_terms_sorted_seq ()

  Compact a presorted expairseq by combining all matching expairs to one each.
- bool is_canonical () const

  Check if this expairseq is in sorted (canonical) form.
- epvector expandchildren (unsigned options) const

  Member-wise expand the expairs in this sequence.
- epvector evalchildren () const

  Member-wise evaluate the expairs in this sequence.
- epvector subschildren (const exmap &m, unsigned options=0) const

  Member-wise substitute in this sequence.

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Static Protected Member Functions

- static bool can_be_further_expanded (const ex &e)

Friends

- class add
- class ncmul
- class power

Additional Inherited Members

Protected Attributes inherited from GiNaC::expairseq

- epvector seq
- ex overall_coeff

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.95.1 Detailed Description

Product of expressions.

8.95.2 Constructor & Destructor Documentation

8.95.2.1 mul() [1/7]

```
GiNaC::mul::mul (
            const ex & lh,
            const ex & rh)
```

References GiNaC::_ex1, GiNaC::expairseq::construct_from_2_ex(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

Referenced by GiNaC::matrix::charpoly(), GiNaC::expairseq::combine_overall_coeff(), GiNaC::Gt_detail::deconcatenate_path(), GiNaC::dirac_trace(), do_print_latex(), expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), GiNaC::Gt_detail::TransformExpressionWithCache< Type >::impl(), GiNaC::log_expand(), power, GiNaC::indexed::simplify_indexed_smod(), and GiNaC::sqrfree().

8.95.2.2 mul() [2/7]

```
GiNaC::mul::mul (
            const exvector & v)
```

References GiNaC::_ex1, GiNaC::expairseq::construct_from_exvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.95.2.3 mul() [3/7]

```
GiNaC::mul::mul (
            const epvector & v)
```

References GiNaC::_ex1, GiNaC::expairseq::construct_from_epvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.95.2.4 mul() [4/7]

```
GiNaC::mul::mul (
            const epvector & v,
            const ex & oc,
            bool do_index_renaming = false)
```

References GiNaC::expairseq::construct_from_epvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.95.2.5 mul() [5/7]

```
GiNaC::mul::mul (
            epvector && vp)
```

References GiNaC::_ex1, GiNaC::expairseq::construct_from_epvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.95.2.6 mul() [6/7]

```
GiNaC::mul::mul (
            epvector && vp,
            const ex & oc,
            bool do_index_renaming = false)
```

References GiNaC::expairseq::construct_from_epvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.95.2.7 mul() [7/7]

```
GiNaC::mul::mul (
            const ex & lh,
            const ex & mh,
            const ex & rh)
```

References GiNaC::_ex1, GiNaC::expairseq::construct_from_exvector(), GINAC_ASSERT, GiNaC::expairseq::is_canonical(), and GiNaC::expairseq::overall_coeff.

8.95.3 Member Function Documentation

8.95.3.1 precedence()

```
unsigned GiNaC::mul::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::expairseq.

References precedence().

Referenced by do_print(), do_print_csrc(), do_print_latex(), precedence(), and print_overall_coeff().

8.95.3.2 info()

```
bool GiNaC::mul::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::expairseq.

References GiNaC::_num1_p, GiNaC::info_flags::cinteger, GiNaC::info_flags::cinteger_polynomial, GiNaC::info_flags::crational, GiNaC::info_flags::crational_polynomial, GiNaC::info_flags::even, GiNaC::factor(), GiNaC::basic::flags, GiNaC::info_flags::indefinite, GiNaC::ex::info(), info(), GiNaC::info_flags::integer, GiNaC::info_flags::integer_polynomial, GiNaC::ex::is_equal(), GiNaC::status_flags::is_negative, GiNaC::status_flags::is_positive, GiNaC::info_flags::negative, GiNaC::info_flags::negint, GiNaC::info_flags::nonnegative, GiNaC::info_flags::nonnegint, GiNaC::expairseq::overall_coeff, GiNaC::info_flags::polynomial, GiNaC::info_flags::posint, GiNaC::info_flags::positive, GiNaC::status_flags::purely_indefinite, GiNaC::info_flags::rational, GiNaC::info_flags::rational_function, GiNaC::info_flags::rational_polynomial, GiNaC::info_flags::real, recombine_pair_to_ex(), GiNaC::expairseq::seq, and GiNaC::basic::setflag().

Referenced by expand(), GiNaC::power::expand_mul(), and info().

8.95.3.3 is_polynomial()

```
bool GiNaC::mul::is_polynomial (
            const ex & var) const  [override], [virtual]
```

Check whether this is a polynomial in the given variables.

Reimplemented from GiNaC::basic.

References GiNaC::info_flags::nonnegint, and GiNaC::expairseq::seq.

8.95.3.4 degree()

```
int GiNaC::mul::degree (
            const ex & s) const  [override], [virtual]
```

Return degree of highest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::ex::degree(), GiNaC::ex_to(), recombine_pair_to_ex(), and GiNaC::expairseq::seq.

8.95.3.5 ldegree()

```
int GiNaC::mul::ldegree (
            const ex & s) const  [override], [virtual]
```

Return degree of lowest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::ex_to(), GiNaC::ex::ldegree(), recombine_pair_to_ex(), and GiNaC::expairseq::seq.

8.95.3.6 coeff()

```
ex GiNaC::mul::coeff (
            const ex & s,
            int n = 1) const  [override], [virtual]
```

Return coefficient of degree n in object s.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::ex::coeff(), coeff(), GiNaC::dynallocate(), GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), and GiNaC::expairseq::seq.

Referenced by coeff(), and print_overall_coeff().

8.95.3.7 has()

```
bool GiNaC::mul::has (
            const ex & pattern,
            unsigned options = 0) const  [override], [virtual]
```

Test for occurrence of a pattern.

An object 'has' a pattern if it matches the pattern itself or one of the children 'has' it. As a consequence (according to the definition of children) given e=x+y+z, e.has(x) is true but e.has(x+y) is false.

Reimplemented from GiNaC::basic.

References GiNaC::has_options::algebraic, GiNaC::algebraic_match_mul_with_mul(), GiNaC::basic::has(), GiNaC::is_a(), and GiNaC::expairseq::nops().

8.95.3.8 eval()

```
ex GiNaC::mul::eval () const  [override], [virtual]
```

Perform automatic term rewriting rules in this class.

In the following x, x1, x2,... stand for a symbolic variables of type ex and c, c1, c2... stand for such expressions that contain a plain number.

- *(...,x;0) -> 0*
- ∗*(+(x1,x2,...);c)* -> ∗*(+(∗(x1,c),*(x2,c),...))
- ∗(x;1) -> x
- ∗(;c) -> c

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::_num1_p, GiNaC::_num_1_p, add, GiNaC::basic::clearflag(), GiNaC::ex::coeff(), GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::numeric::div(), GiNaC::dynallocate(), GiNaC::expairseq::evalchildren(), GiNaC::status_flags::evaluated, GiNaC::ex_to(), GiNaC::expair::expair(), GiNaC::status_flags::expanded, GiNaC::basic::flags, GINAC_ASSERT, GiNaC::status_flags::hash_calculated, GiNaC::basic::hold(), GiNaC::numeric::integer_content(), GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_pos_integer(), GiNaC::ex::is_zero(), likely, GiNaC::numeric::mul(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), GiNaC::expairseq::seq, and unlikely.

8.95.3.9 evalf()

```
ex GiNaC::mul::evalf () const  [override], [virtual]
```

Evaluate object numerically.

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate(), GiNaC::ex::evalf(), GiNaC::expair::expair(), GiNaC::expairseq::overall_coeff, and GiNaC::expairseq::seq.

8.95.3.10 real_part()

```
ex GiNaC::mul::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References find_real_imag().

8.95.3.11 imag_part()

```
ex GiNaC::mul::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References find_real_imag().

8.95.3.12 evalm()

```
ex GiNaC::mul::evalm () const  [override], [virtual]
```

Evaluate sums, products and integer powers of matrices.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::dynallocate(), GiNaC::ex::evalm(), GiNaC::ex_to(), GiNaC::is_a(), GiNaC::matrix::mul_scalar(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), GiNaC::expairseq::seq, and split_ex_to_pair().

8.95.3.13 series()

```
ex GiNaC::mul::series (
            const relational & r,
            int order,
            unsigned options = 0) const  [override], [virtual]
```

Implementation of ex::series() for product.

This performs series multiplication when multiplying series.

See also

ex::series

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::ex::coeff(), GiNaC::dynallocate(), GiNaC::ex_to(), GiNaC::ex::expand(), GiNaC::factor(), GINAC_ASSERT, GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::is_a(), GiNaC::ex::ldegree(), GiNaC::relational::lhs(), GiNaC::pseries::mul_const(), GiNaC::pseries::mul_series(), GiNaC::expairseq::op(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), GiNaC::relational::rhs(), GiNaC::expairseq::seq, and GiNaC::ex::series().

8.95.3.14 normal()

```
ex GiNaC::mul::normal (
            exmap & repl,
            exmap & rev_lookup,
            lst & modifier) const  [override], [virtual]
```

Implementation of ex::normal() for a product.

It cancels common factors from fractions.

See also

ex::normal()

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate(), GiNaC::ex_to(), GiNaC::frac_cancel(), GiNaC::subs_options::no_pattern, GiNaC::container< C >::nops(), GiNaC::ex::normal(), GiNaC::container< C >::op(), GiNaC::ex::op(), GiNaC::expairseq::overall_coef recombine_pair_to_ex(), and GiNaC::expairseq::seq.

8.95.3.15 integer_content()

```
numeric GiNaC::mul::integer_content () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::abs(), GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), and GiNaC::expairseq::seq.

8.95.3.16 smod()

```
ex GiNaC::mul::smod (
            const numeric & xi) const  [override], [virtual]
```

Apply symmetric modular homomorphism to an expanded multivariate polynomial.

This function is usually used internally by heur_gcd().

Parameters

xi	modulus

Returns

mapped polynomial

See also

heur_gcd

Reimplemented from GiNaC::basic.

References GiNaC::basic::clearflag(), GiNaC::dynallocate(), GiNaC::status_flags::evaluated, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::status_flags::hash_calculated, GiNaC::is_exactly_a(), mul(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), GiNaC::expairseq::seq, and GiNaC::smod().

8.95.3.17 max_coefficient()

```
numeric GiNaC::mul::max_coefficient () const  [override], [virtual]
```

Implementation ex::max_coefficient().

See also

heur_gcd

Reimplemented from GiNaC::basic.

References GiNaC::abs(), GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), and GiNaC::expairseq::seq.

8.95.3.18 get_free_indices()

```
exvector GiNaC::mul::get_free_indices () const  [override], [virtual]
```

Return a vector containing the free indices of an expression.

Reimplemented from GiNaC::basic.

References GiNaC::find_free_and_dummy(), GiNaC::ex::get_free_indices(), GiNaC::expairseq::nops(), and GiNaC::expairseq::op().

8.95.3.19 conjugate()

```
ex GiNaC::mul::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::are_ex_trivially_equal(), GiNaC::ex::conjugate(), GiNaC::ex::is_equal(), GiNaC::expairseq::overall_coeff, recombine_pair_to_ex(), GiNaC::expairseq::seq, split_ex_to_pair(), and thisexpairseq().

8.95.3.20 derivative()

```
ex GiNaC::mul::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for a product.

It applies the product rule.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::ex::coeff(), GiNaC::basic::diff(), GiNaC::dynallocate(), GiNaC::expairseq::overall_coeff, GiNaC::pow(), GiNaC::expairseq::seq, split_ex_to_pair(), and GiNaC::expair::swap().

8.95.3.21 eval_ncmul()

```
ex GiNaC::mul::eval_ncmul (
            const exvector & v) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::noncommutative, and GiNaC::expairseq::seq.

8.95.3.22 return_type()

```
unsigned GiNaC::mul::return_type () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::return_types::commutative, GiNaC::return_types::noncommutative, GiNaC::return_types::noncommutative_comp
and GiNaC::expairseq::seq.

8.95.3.23 return_type_tinfo()

```
return_type_t GiNaC::mul::return_type_tinfo () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::make_return_type_t(), GiNaC::return_types::noncommutative, and GiNaC::expairseq::seq.

8.95.3.24 thisexpairseq() [1/2]

```
ex GiNaC::mul::thisexpairseq (
            const epvector & v,
            const ex & oc,
            bool do_index_renaming = false) const  [override], [protected], [virtual]
```

Create an object of this type.

This method works similar to a constructor. It is useful because expairseq has (at least) two possible different semantics but we want to inherit methods thus avoiding code duplication. Sometimes a method in expairseq has to create a new one of the same semantics, which cannot be done by a ctor because the name (add, mul,...) is unknown on the expairseq level. In order for this trick to work a derived class must of course override this definition.

Reimplemented from GiNaC::expairseq.

References GiNaC::dynallocate().

Referenced by conjugate().

8.95.3.25 thisexpairseq() [2/2]

```
ex GiNaC::mul::thisexpairseq (
            epvector && vp,
            const ex & oc,
            bool do_index_renaming = false) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::dynallocate().

8.95.3.26 split_ex_to_pair()

```
expair GiNaC::mul::split_ex_to_pair (
            const ex & e) const  [override], [protected], [virtual]
```

Form an expair from an ex, using the corresponding semantics.

See also

expairseq::recombine_pair_to_ex()

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1, GiNaC::power::basis, GiNaC::ex_to(), GiNaC::expair::expair(), GiNaC::power::exponent, GiNaC::is_exactly_a(), and power.

Referenced by combine_ex_with_coeff_to_pair(), combine_pair_with_coeff_to_pair(), conjugate(), derivative(), evalm(), expair_needs_further_processing(), expand(), and expandchildren().

8.95.3.27 combine_ex_with_coeff_to_pair()

```
expair GiNaC::mul::combine_ex_with_coeff_to_pair (
            const ex & e,
            const ex & c) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1, GiNaC::expair::expair(), GINAC_ASSERT, GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::pow(), and split_ex_to_pair().

8.95.3.28 combine_pair_with_coeff_to_pair()

```
expair GiNaC::mul::combine_pair_with_coeff_to_pair (
            const expair & p,
            const ex & c) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1, GiNaC::expair::coeff, GiNaC::expair::expair(), GINAC_ASSERT, GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::pow(), recombine_pair_to_ex(), GiNaC::expair::rest, and split_ex_to_pair().

Referenced by GiNaC::power::expand_mul().

8.95.3.29 recombine_pair_to_ex()

```
ex GiNaC::mul::recombine_pair_to_ex (
            const expair & p) const  [override], [protected], [virtual]
```

Form an ex out of an expair, using the corresponding semantics.

See also

expairseq::split_ex_to_pair()

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1, GiNaC::expair::coeff, GiNaC::dynallocate(), GiNaC::ex::is_equal(), and GiNaC::expair::rest.

Referenced by coeff(), combine_pair_with_coeff_to_pair(), conjugate(), degree(), do_print(), do_print_latex(), eval(), evalm(), expair_needs_further_processing(), GiNaC::power::expand(), expandchildren(), find_real_imag(), info(), integer_content(), ldegree(), max_coefficient(), normal(), series(), and smod().

8.95.3.30 expair_needs_further_processing()

```
bool GiNaC::mul::expair_needs_further_processing (
            epp it)  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1, GiNaC::ex_to(), GiNaC::expair::is_equal(), GiNaC::is_exactly_a(), recombine_pair_to_ex(), and split_ex_to_pair().

8.95.3.31 default_overall_coeff()

```
ex GiNaC::mul::default_overall_coeff () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1.

8.95.3.32 combine_overall_coeff() [1/2]

```
void GiNaC::mul::combine_overall_coeff (
            const ex & c)  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), and GiNaC::expairseq::overall_coeff.

8.95.3.33 combine_overall_coeff() [2/2]

```
void GiNaC::mul::combine_overall_coeff (
            const ex & c1,
            const ex & c2)  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::expairseq::overall_coeff, and GiNaC::power::power().

8.95.3.34 can_make_flat()

```
bool GiNaC::mul::can_make_flat (
            const expair & p) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::expairseq.

References GiNaC::expair::coeff, GINAC_ASSERT, GiNaC::ex::info(), GiNaC::info_flags::integer, and GiNaC::is_exactly_a().

8.95.3.35 expand()

```
ex GiNaC::mul::expand (
            unsigned options = 0) const  [override], [protected], [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented from GiNaC::expairseq.

References GiNaC::_ex1, GiNaC::_num0_p, add, can_be_further_expanded(), GiNaC::dynallocate(), GiNaC::ex_to(), GiNaC::expair::expair(), GiNaC::ex::expand(), GiNaC::expand_options::expand_rename_idx, expandchildren(), GiNaC::status_flags::expanded, GiNaC::get_all_dummy_indices_safely(), GiNaC::info_flags::has_indices, info(), GiNaC::info_flags::integer, GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::ex::is_zero(), mul(), GiNaC::numeric::mul(), GiNaC::subs_options::no_pattern, GiNaC::ex::nops(), GiNaC::expairseq::nops(), GiNaC::container< C >::op(), GiNaC::ex::op(), GiNaC::expairseq::overall_coeff, GiNaC::rename_dummy_indices_uniquely(), GiNaC::expairseq::seq, GiNaC::basic::setflag(), and split_ex_to_pair().

8.95.3.36 algebraic_subs_mul()

```
ex GiNaC::mul::algebraic_subs_mul (
            const exmap & m,
            unsigned options) const
```

References GiNaC::subs_options::algebraic, GiNaC::algebraic_match_mul_with_mul(), GiNaC::is_exactly_a(), GiNaC::subs_options::no_pattern, GiNaC::expairseq::nops(), GiNaC::expairseq::op(), GiNaC::pow(), GiNaC::ex::subs(), GiNaC::basic::subs_one_level(), and GiNaC::tryfactsubs().

8.95.3.37 find_real_imag()

```
void GiNaC::mul::find_real_imag (
            ex & rp,
            ex & ip) const  [protected]
```

References GiNaC::ex::expand(), GiNaC::factor(), GiNaC::ex::imag_part(), GiNaC::ex::is_zero(), GiNaC::expairseq::overall_coeff, GiNaC::ex::real_part(), recombine_pair_to_ex(), and GiNaC::expairseq::seq.

Referenced by imag_part(), and real_part().

8.95.3.38 print_overall_coeff()

```
void GiNaC::mul::print_overall_coeff (
            const print_context & c,
            const char * mul_sym) const  [protected]
```

References GiNaC::_num1_p, GiNaC::_num_1_p, coeff(), GiNaC::ex_to(), GiNaC::ex::is_equal(), GiNaC::expairseq::overall_coeff, precedence(), GiNaC::basic::print(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by do_print(), and do_print_latex().

8.95.3.39 do_print()

```
void GiNaC::mul::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References GiNaC::class_info< OPT >::first, precedence(), GiNaC::ex::print(), print_overall_coeff(), recombine_pair_to_ex(), GiNaC::print_context::s, and GiNaC::expairseq::seq.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.95.3.40 do_print_latex()

```
void GiNaC::mul::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

References GiNaC::ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, GiNaC::is_exactly_a(), mul(), precedence(), print_overall_coeff(), recombine_pair_to_ex(), GiNaC::print_context::s, and GiNaC::expairseq::seq.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.95.3.41 do_print_csrc()

```
void GiNaC::mul::do_print_csrc (
            const print_csrc & c,
            unsigned level) const  [protected]
```

References GiNaC::_ex1, GiNaC::_ex_1, GiNaC::ex::ex(), GiNaC::ex_to(), GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::info_flags::negint, GiNaC::expairseq::overall_coeff, GiNaC::power::power(), precedence(), GiNaC::ex::print(), GiNaC::print_context::s, and GiNaC::expairseq::seq.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.95.3.42 do_print_python_repr()

```
void GiNaC::mul::do_print_python_repr (
            const print_python_repr & c,
            unsigned level) const  [protected]
```

References GiNaC::expairseq::nops(), GiNaC::expairseq::op(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.95.3.43 can_be_further_expanded()

```
bool GiNaC::mul::can_be_further_expanded (
            const ex & e)  [static], [protected]
```

References GiNaC::ex_to(), GiNaC::ex::info(), GiNaC::is_exactly_a(), GiNaC::ex::op(), GiNaC::info_flags::posint, and GiNaC::expairseq::seq.

Referenced by expand().

8.95.3.44 expandchildren()

```
epvector GiNaC::mul::expandchildren (
            unsigned options) const  [protected]
```

Member-wise expand the expairs representing this sequence.

This must be overridden from expairseq::expandchildren() and done iteratively in order to allow for early cancellations and thus safe memory.

See also

mul::expand()

Returns

epvector containing expanded pairs, empty if no members had to be changed.

References GiNaC::are_ex_trivially_equal(), GiNaC::ex::expand(), GiNaC::factor(), recombine_pair_to_ex(), GiNaC::expairseq::seq, and split_ex_to_pair().

Referenced by expand().

8.95.4 Friends And Related Symbol Documentation

8.95.4.1 add

```
friend class add  [friend]
```

References add.

Referenced by add, eval(), and expand().

8.95.4.2 ncmul

```
friend class ncmul  [friend]
```

References ncmul.

Referenced by ncmul.

8.95.4.3 power

```
friend class power  [friend]
```

References mul(), and power.

Referenced by power, and split_ex_to_pair().

The documentation for this class was generated from the following files:

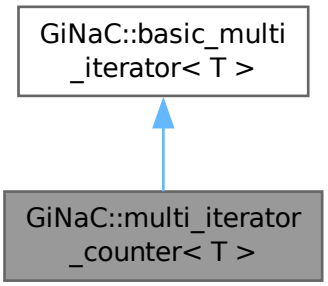
- mul.h
- indexed.cpp
- mul.cpp
- normal.cpp
- pseries.cpp

8.96 GiNaC::multi_iterator_counter< T > Class Template Reference

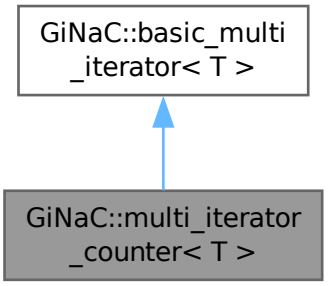
The class multi_iterator_counter defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

`#include <utils_multi_iterator.h>`

Inheritance diagram for GiNaC::multi_iterator_counter< T >:



Collaboration diagram for GiNaC::multi_iterator_counter< T >:



Public Member Functions

- multi_iterator_counter (void)
  Default constructor.
- multi_iterator_counter (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N and size k .
- multi_iterator_counter (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- basic_multi_iterator< T > & init (void)
  Initialize the multi-index to.
- basic_multi_iterator< T > & operator++ (int)
  The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

Public Member Functions inherited from GiNaC::basic_multi_iterator< T >

- basic_multi_iterator (void)
  Default constructor.
- basic_multi_iterator (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N, lower limit B and size k .
- basic_multi_iterator (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- virtual ~basic_multi_iterator ()
  Destructor.
- size_t size (void) const
  Returns the size of a multi_iterator.
- bool overflow (void) const
  Return the overflow flag.
- const std::vector< T > & get_vector (void) const
  Returns a reference to the vector v.
- T operator[ ] (size_t i) const
  Subscription via [].
- T & operator[ ] (size_t i)
  Subscription via [].
- T operator() (size_t i) const
  Subscription via ().
- T & operator() (size_t i)
  Subscription via ().

Friends

- template<class TT>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_counter< TT > &v)

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic_multi_iterator< T >

- T N
- T B
- std::vector< T > v
- bool flag_overflow

8.96.1 Detailed Description

template<class T>
class GiNaC::multi_iterator_counter< T >

The class multi_iterator_counter defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

$$B \leq i_j < N$$

8.96.2 Constructor & Destructor Documentation

8.96.2.1 multi_iterator_counter() [1/3]

```
template<class T>
GiNaC::multi_iterator_counter< T >::multi_iterator_counter (
            void )  [inline]
```

Default constructor.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator().

Referenced by operator<<.

8.96.2.2 multi_iterator_counter() [2/3]

```
template<class T>
GiNaC::multi_iterator_counter< T >::multi_iterator_counter (
            T B,
            T N,
            size_t k)  [inline], [explicit]
```

Construct a multi_iterator with upper limit N and size k .

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), and GiNaC::basic_multi_iterator< T >::N.

8.96.2.3 multi_iterator_counter() [3/3]

```
template<class T>
GiNaC::multi_iterator_counter< T >::multi_iterator_counter (
            T B,
            T N,
            const std::vector< T > & vv)  [inline], [explicit]
```

Construct from a vector.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::N, and GiNaC::basic_multi_iterator< T >::v.

8.96.3 Member Function Documentation

8.96.3.1 init()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_counter< T >::init (
            void )  [inline], [virtual]
```

Initialize the multi-index to.

$$(n_1, n_2, n_3, ..., n_k) = (B, B, ..., B)$$

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, and GiNaC::basic_multi_iterator< T >::v.

8.96.3.2 operator++()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_counter< T >::operator++ (
            int )  [inline], [virtual]
```

The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

If n is in the last configuration and the increment operator ++ is applied to n, the overflow flag will be raised.

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, GiNaC::basic_multi_iterator< T >::N, GiNaC::basic_multi_iterator< T >::size(), and GiNaC::basic_multi_iterator< T >::v.

8.96.4 Friends And Related Symbol Documentation

8.96.4.1 operator<<

```
template<class T>
template<class TT>
std::ostream & operator<< (
            std::ostream & os,
            const multi_iterator_counter< TT > & v)  [friend]
```

References multi_iterator_counter(), and GiNaC::basic_multi_iterator< T >::v.

The documentation for this class was generated from the following file:

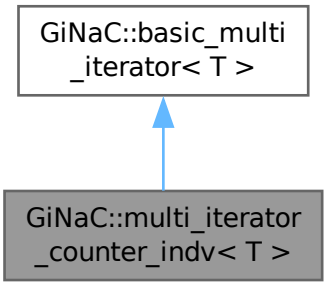
- utils_multi_iterator.h

8.97 GiNaC::multi_iterator_counter_indv< T > Class Template Reference

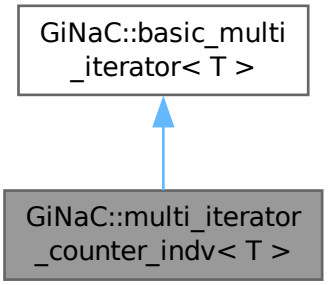
The class multi_iterator_counter_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

`#include <utils_multi_iterator.h>`

Inheritance diagram for GiNaC::multi_iterator_counter_indv< T >:



Collaboration diagram for GiNaC::multi_iterator_counter_indv< T >:



Public Member Functions

- multi_iterator_counter_indv (void)
  Default constructor.
- multi_iterator_counter_indv (T B, const std::vector< T > &Nv, size_t k)
  Construct a multi_iterator with upper limit N and size k .
- multi_iterator_counter_indv (T B, const std::vector< T > &Nv, const std::vector< T > &vv)
  Construct from a vector.
- basic_multi_iterator< T > & init (void)
  Initialize the multi-index to.
- basic_multi_iterator< T > & operator++ (int)
  The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

Public Member Functions inherited from GiNaC::basic_multi_iterator< T >

- basic_multi_iterator (void)
  Default constructor.
- basic_multi_iterator (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N, lower limit B and size k .
- basic_multi_iterator (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- virtual ~basic_multi_iterator ()
  Destructor.
- size_t size (void) const
  Returns the size of a multi_iterator.
- bool overflow (void) const
  Return the overflow flag.
- const std::vector< T > & get_vector (void) const
  Returns a reference to the vector v.
- T operator[ ] (size_t i) const
  Subscription via [].
- T & operator[ ] (size_t i)

Subscription via [].

- T operator() (size_t i) const

  Subscription via ().

- T & operator() (size_t i)

  Subscription via ().

Protected Attributes

- std::vector< T > Nv

Protected Attributes inherited from GiNaC::basic_multi_iterator< T >

- T N
- T B
- std::vector< T > v
- bool flag_overflow

Friends

- template<class TT>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_counter_indv< TT > &v)

8.97.1 Detailed Description

template<class T>
class GiNaC::multi_iterator_counter_indv< T >

The class multi_iterator_counter_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

$$B \leq i_j < N_j$$

8.97.2 Constructor & Destructor Documentation

8.97.2.1 multi_iterator_counter_indv() [1/3]

```
template<class T>
GiNaC::multi_iterator_counter_indv< T >::multi_iterator_counter_indv (
            void )  [inline]
```

Default constructor.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), and Nv.

Referenced by operator<<.

8.97.2.2 multi_iterator_counter_indv() [2/3]

```
template<class T>
GiNaC::multi_iterator_counter_indv< T >::multi_iterator_counter_indv (
            T B,
            const std::vector< T > & Nv,
            size_t k)  [inline], [explicit]
```

Construct a multi_iterator with upper limit N and size k .

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), and Nv.

8.97.2.3 multi_iterator_counter_indv() [3/3]

```
template<class T>
GiNaC::multi_iterator_counter_indv< T >::multi_iterator_counter_indv (
            T B,
            const std::vector< T > & Nv,
            const std::vector< T > & vv)  [inline], [explicit]
```

Construct from a vector.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), Nv, and GiNaC::basic_multi_iterator< T >::v.

8.97.3 Member Function Documentation

8.97.3.1 init()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_counter_indv< T >::init (
            void )  [inline], [virtual]
```

Initialize the multi-index to.

$$(n_1, n_2, n_3, ..., n_k) = (B, B, ..., B)$$

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, and GiNaC::basic_multi_iterator< T >::v.

8.97.3.2 operator++()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_counter_indv< T >::operator++ (
            int )  [inline], [virtual]
```

The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

If n is in the last configuration and the increment operator ++ is applied to n, the overflow flag will be raised.

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, Nv, GiNaC::basic_multi_iterator< T >::size(), and GiNaC::basic_multi_iterator< T

8.97.4 Friends And Related Symbol Documentation

8.97.4.1 operator<<

```
template<class T>
template<class TT>
std::ostream & operator<< (
            std::ostream & os,
            const multi_iterator_counter_indv< TT > & v)  [friend]
```

References multi_iterator_counter_indv(), and GiNaC::basic_multi_iterator< T >::v.

8.97.5 Member Data Documentation

8.97.5.1 Nv

```
template<class T>
std::vector<T> GiNaC::multi_iterator_counter_indv< T >::Nv  [protected]
```

Referenced by multi_iterator_counter_indv(), multi_iterator_counter_indv(), multi_iterator_counter_indv(), and operator++().

The documentation for this class was generated from the following file:

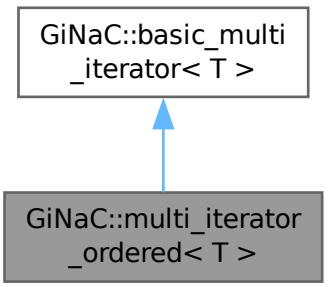
- utils_multi_iterator.h

8.98 GiNaC::multi_iterator_ordered< T > Class Template Reference

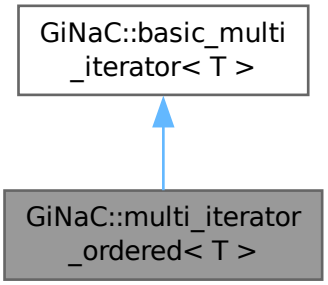
The class multi_iterator_ordered defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

```
#include <utils_multi_iterator.h>
```

Inheritance diagram for GiNaC::multi_iterator_ordered< T >:



Collaboration diagram for GiNaC::multi_iterator_ordered< T >:



Public Member Functions

- multi_iterator_ordered (void)
  Default constructor.
- multi_iterator_ordered (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N and size k .
- multi_iterator_ordered (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- basic_multi_iterator< T > & init (void)
  Initialize the multi-index to.
- basic_multi_iterator< T > & operator++ (int)
  The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

Public Member Functions inherited from GiNaC::basic_multi_iterator< T >

- basic_multi_iterator (void)
  Default constructor.
- basic_multi_iterator (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N, lower limit B and size k .
- basic_multi_iterator (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- virtual ~basic_multi_iterator ()
  Destructor.
- size_t size (void) const
  Returns the size of a multi_iterator.
- bool overflow (void) const
  Return the overflow flag.
- const std::vector< T > & get_vector (void) const
  Returns a reference to the vector v.
- T operator[ ] (size_t i) const
  Subscription via [].
- T & operator[ ] (size_t i)

Subscription via [].

- T operator() (size_t i) const

  Subscription via ().

- T & operator() (size_t i)

  Subscription via ().

Friends

- template<class TT>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_ordered< TT > &v)

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic_multi_iterator< T >

- T N
- T B
- std::vector< T > v
- bool flag_overflow

8.98.1 Detailed Description

template<class T>
class GiNaC::multi_iterator_ordered< T >

The class multi_iterator_ordered defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

$$B \leq i_j < N$$

and

$$i_j < i_{j+1}.$$

It is assumed that $k > 0$ and $N - B \geq k$.

8.98.2 Constructor & Destructor Documentation

8.98.2.1 multi_iterator_ordered() [1/3]

```
template<class T>
GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered (
            void )  [inline]
```

Default constructor.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator().

Referenced by operator<<.

8.98.2.2 multi_iterator_ordered() [2/3]

```
template<class T>
GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered (
            T B,
            T N,
            size_t k)  [inline], [explicit]
```

Construct a multi_iterator with upper limit N and size k .

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), and GiNaC::basic_multi_iterator< T >::N.

8.98.2.3 multi_iterator_ordered() [3/3]

```
template<class T>
GiNaC::multi_iterator_ordered< T >::multi_iterator_ordered (
            T B,
            T N,
            const std::vector< T > & vv)  [inline], [explicit]
```

Construct from a vector.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::N, and GiNaC::basic_multi_iterator< T >::v.

8.98.3 Member Function Documentation

8.98.3.1 init()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_ordered< T >::init (
            void )  [inline], [virtual]
```

Initialize the multi-index to.

$$(n_1, n_2, n_3, ..., n_k) = (B + 0, B + 1, B + 2, ..., B + k - 1)$$

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, and GiNaC::basic_multi_iterator< T >::v.

8.98.3.2 operator++()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_ordered< T >::operator++ (
            int )  [inline], [virtual]
```

The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

If n is in the last configuration and the increment operator ++ is applied to n, the overflow flag will be raised.

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, GiNaC::basic_multi_iterator< T >::N, GiNaC::basic_multi_iterator< T >::size(), and GiNaC::basic_multi_iterator< T >::v.

8.98.4 Friends And Related Symbol Documentation

8.98.4.1 operator<<

```
template<class T>
template<class TT>
std::ostream & operator<< (
            std::ostream & os,
            const multi_iterator_ordered< TT > & v)  [friend]
```

References multi_iterator_ordered(), and GiNaC::basic_multi_iterator< T >::v.

The documentation for this class was generated from the following file:

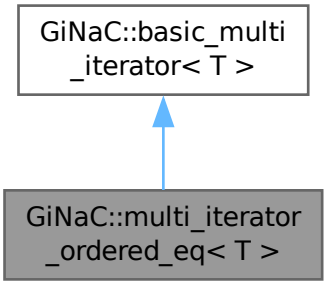
- utils_multi_iterator.h

8.99 GiNaC::multi_iterator_ordered_eq< T > Class Template Reference

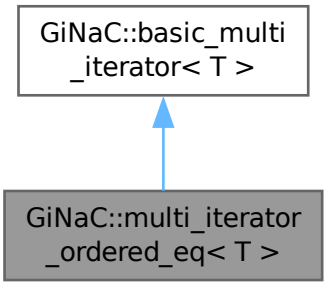
The class multi_iterator_ordered_eq defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

`#include <utils_multi_iterator.h>`

Inheritance diagram for GiNaC::multi_iterator_ordered_eq< T >:



Collaboration diagram for GiNaC::multi_iterator_ordered_eq< T >:



Public Member Functions

- multi_iterator_ordered_eq (void)
  Default constructor.
- multi_iterator_ordered_eq (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N and size k .
- multi_iterator_ordered_eq (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- basic_multi_iterator< T > & init (void)
  Initialize the multi-index to.
- basic_multi_iterator< T > & operator++ (int)
  The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

Public Member Functions inherited from GiNaC::basic_multi_iterator< T >

- basic_multi_iterator (void)
  Default constructor.
- basic_multi_iterator (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N, lower limit B and size k .
- basic_multi_iterator (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- virtual ~basic_multi_iterator ()
  Destructor.
- size_t size (void) const
  Returns the size of a multi_iterator.
- bool overflow (void) const
  Return the overflow flag.
- const std::vector< T > & get_vector (void) const
  Returns a reference to the vector v.
- T operator[ ] (size_t i) const
  Subscription via [].
- T & operator[ ] (size_t i)
  Subscription via [].
- T operator() (size_t i) const
  Subscription via ().
- T & operator() (size_t i)
  Subscription via ().

Friends

- template<class TT>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_ordered_eq< TT > &v)

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic_multi_iterator< T >

- T N
- T B
- std::vector< T > v
- bool flag_overflow

8.99.1 Detailed Description

template<class T>
class GiNaC::multi_iterator_ordered_eq< T >

The class multi_iterator_ordered_eq defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

$$B \leq i_j < N$$

and

$$i_j \leq i_{j+1}.$$

It is assumed that $k > 0$.

8.99.2 Constructor & Destructor Documentation

8.99.2.1 multi_iterator_ordered_eq() [1/3]

```
template<class T>
GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq (
            void )  [inline]
```

Default constructor.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator().

Referenced by operator<<.

8.99.2.2 multi_iterator_ordered_eq() [2/3]

```
template<class T>
GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq (
            T B,
            T N,
            size_t k)  [inline], [explicit]
```

Construct a multi_iterator with upper limit N and size k .

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), and GiNaC::basic_multi_iterator< T >::N.

8.99.2.3 multi_iterator_ordered_eq() [3/3]

```
template<class T>
GiNaC::multi_iterator_ordered_eq< T >::multi_iterator_ordered_eq (
            T B,
            T N,
            const std::vector< T > & vv)  [inline], [explicit]
```

Construct from a vector.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::N, and GiNaC::basic_multi_iterator< T >::v.

8.99.3 Member Function Documentation

8.99.3.1 init()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_ordered_eq< T >::init (
            void )  [inline], [virtual]
```

Initialize the multi-index to.

$$(n_1, n_2, ..., n_k) = (B, B, ..., B)$$

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, and GiNaC::basic_multi_iterator< T >::v.

8.99.3.2 operator++()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_ordered_eq< T >::operator++ (
            int )  [inline], [virtual]
```

The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

If n is in the last configuration and the increment operator ++ is applied to n, the overflow flag will be raised.

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, GiNaC::basic_multi_iterator< T >::N, GiNaC::basic_multi_iterator< T >::size(), and GiNaC::basic_multi_iterator< T >::v.

8.99.4 Friends And Related Symbol Documentation

8.99.4.1 operator<<

```
template<class T>
template<class TT>
std::ostream & operator<< (
            std::ostream & os,
            const multi_iterator_ordered_eq< TT > & v)  [friend]
```

References multi_iterator_ordered_eq(), and GiNaC::basic_multi_iterator< T >::v.

The documentation for this class was generated from the following file:

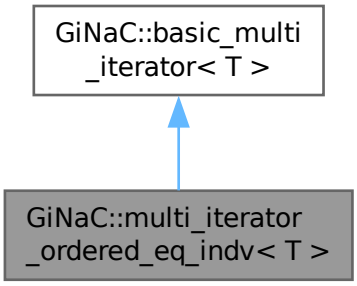
- utils_multi_iterator.h

8.100 GiNaC::multi_iterator_ordered_eq_indv< T > Class Template Reference

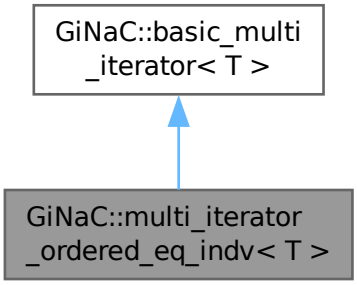
The class multi_iterator_ordered_eq_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

`#include <utils_multi_iterator.h>`

Inheritance diagram for GiNaC::multi_iterator_ordered_eq_indv< T >:



Collaboration diagram for GiNaC::multi_iterator_ordered_eq_indv< T >:



Public Member Functions

- multi_iterator_ordered_eq_indv (void)
  Default constructor.
- multi_iterator_ordered_eq_indv (T B, const std::vector< T > &Nv, size_t k)
  Construct a multi_iterator with upper limit N and size k .
- multi_iterator_ordered_eq_indv (T B, const std::vector< T > &Nv, const std::vector< T > &vv)
  Construct from a vector.
- basic_multi_iterator< T > & init (void)
  Initialize the multi-index to.
- basic_multi_iterator< T > & operator++ (int)
  The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

Public Member Functions inherited from GiNaC::basic_multi_iterator< T >

- basic_multi_iterator (void)
  Default constructor.
- basic_multi_iterator (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N, lower limit B and size k .
- basic_multi_iterator (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- virtual ~basic_multi_iterator ()
  Destructor.
- size_t size (void) const
  Returns the size of a multi_iterator.
- bool overflow (void) const
  Return the overflow flag.
- const std::vector< T > & get_vector (void) const
  Returns a reference to the vector v.
- T operator[ ] (size_t i) const
  Subscription via [].
- T & operator[ ] (size_t i)
  Subscription via [].
- T operator() (size_t i) const
  Subscription via ().
- T & operator() (size_t i)
  Subscription via ().

Protected Attributes

- std::vector< T > Nv

Protected Attributes inherited from GiNaC::basic_multi_iterator< T >

- T N
- T B
- std::vector< T > v
- bool flag_overflow

Friends

- template<class TT>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_ordered_eq_indv< TT > &v)

8.100.1 Detailed Description

template<class T>
class GiNaC::multi_iterator_ordered_eq_indv< T >

The class multi_iterator_ordered_eq_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

$$B \leq i_j < N_j$$

and

$$i_j \leq i_{j+1}.$$

8.100.2 Constructor & Destructor Documentation

8.100.2.1 multi_iterator_ordered_eq_indv() [1/3]

```
template<class T>
GiNaC::multi_iterator_ordered_eq_indv< T >::multi_iterator_ordered_eq_indv (
            void )  [inline]
```

Default constructor.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), and Nv.

Referenced by operator<<.

8.100.2.2 multi_iterator_ordered_eq_indv() [2/3]

```
template<class T>
GiNaC::multi_iterator_ordered_eq_indv< T >::multi_iterator_ordered_eq_indv (
            T B,
            const std::vector< T > & Nv,
            size_t k)  [inline], [explicit]
```

Construct a multi_iterator with upper limit N and size k .

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), and Nv.

8.100.2.3 multi_iterator_ordered_eq_indv() [3/3]

```
template<class T>
GiNaC::multi_iterator_ordered_eq_indv< T >::multi_iterator_ordered_eq_indv (
            T B,
            const std::vector< T > & Nv,
            const std::vector< T > & vv)  [inline], [explicit]
```

Construct from a vector.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), Nv, and GiNaC::basic_multi_iterator< T >::v.

8.100.3 Member Function Documentation

8.100.3.1 init()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_ordered_eq_indv< T >::init (
            void )  [inline], [virtual]
```

Initialize the multi-index to.

$$(n_1, n_2, n_3, ..., n_k) = (B, B, B, ..., B)$$

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, and GiNaC::basic_multi_iterator< T >::v.

8.100.3.2 operator++()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_ordered_eq_indv< T >::operator++ (
            int )  [inline], [virtual]
```

The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

If n is in the last configuration and the increment operator ++ is applied to n, the overflow flag will be raised.

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, Nv, GiNaC::basic_multi_iterator< T >::size(), and GiNaC::basic_multi_iterator< T >::v.

8.100.4 Friends And Related Symbol Documentation

8.100.4.1 operator<<

```
template<class T>
template<class TT>
std::ostream & operator<< (
            std::ostream & os,
            const multi_iterator_ordered_eq_indv< TT > & v)  [friend]
```

References multi_iterator_ordered_eq_indv(), and GiNaC::basic_multi_iterator< T >::v.

8.100.5 Member Data Documentation

8.100.5.1 Nv

```
template<class T>
std::vector<T> GiNaC::multi_iterator_ordered_eq_indv< T >::Nv  [protected]
```

Referenced by multi_iterator_ordered_eq_indv(), multi_iterator_ordered_eq_indv(), multi_iterator_ordered_eq_indv(), and operator++().

The documentation for this class was generated from the following file:

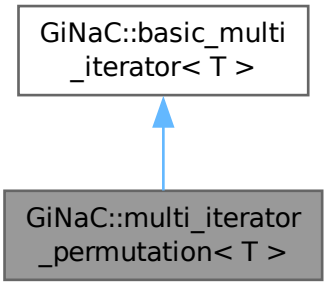
- utils_multi_iterator.h

8.101 GiNaC::multi_iterator_permutation< T > Class Template Reference

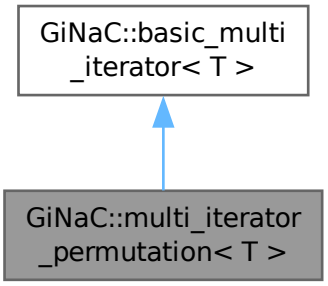
The class multi_iterator_permutation defines a multi_iterator $(i_1, i_2, ..., i_k)$, for which.

`#include <utils_multi_iterator.h>`

Inheritance diagram for GiNaC::multi_iterator_permutation< T >:



Collaboration diagram for GiNaC::multi_iterator_permutation< T >:



Public Member Functions

- multi_iterator_permutation (void)

  Default constructor.
- multi_iterator_permutation (T B, T N, size_t k)

  Construct a multi_iterator with upper limit N and size k .
- multi_iterator_permutation (T B, T N, const std::vector< T > &vv)

  Construct from a vector.

- basic_multi_iterator< T > & init (void)

  Initialize the multi-index to.
- basic_multi_iterator< T > & operator++ (int)

  The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.
- int get_sign (void) const

  Returns the sign of the permutation, defined by.

Public Member Functions inherited from GiNaC::basic_multi_iterator< T >

- basic_multi_iterator (void)

  Default constructor.
- basic_multi_iterator (T B, T N, size_t k)

  Construct a multi_iterator with upper limit N, lower limit B and size k .
- basic_multi_iterator (T B, T N, const std::vector< T > &vv)

  Construct from a vector.
- virtual ~basic_multi_iterator ()

  Destructor.
- size_t size (void) const

  Returns the size of a multi_iterator.
- bool overflow (void) const

  Return the overflow flag.
- const std::vector< T > & get_vector (void) const

  Returns a reference to the vector v.
- T operator[ ] (size_t i) const

  Subscription via [].
- T & operator[ ] (size_t i)

  Subscription via [].
- T operator() (size_t i) const

  Subscription via ().
- T & operator() (size_t i)

  Subscription via ().

Friends

- template<class TT>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_permutation< TT > &v)

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic_multi_iterator< T >

- T N
- T B
- std::vector< T > v
- bool flag_overflow

8.101.1 Detailed Description

template<class T>
class GiNaC::multi_iterator_permutation< T >

The class multi_iterator_permutation defines a multi_iterator $(i_1, i_2, ..., i_k)$, for which.

$$B \leq i_j < N$$

and

$$i_i \neq i_j$$

In particular, if $N - B = k$, multi_iterator_permutation loops over all permutations of k elements.

8.101.2 Constructor & Destructor Documentation

8.101.2.1 multi_iterator_permutation() [1/3]

```
template<class T>
GiNaC::multi_iterator_permutation< T >::multi_iterator_permutation (
            void )  [inline]
```

Default constructor.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator().

Referenced by operator<<.

8.101.2.2 multi_iterator_permutation() [2/3]

```
template<class T>
GiNaC::multi_iterator_permutation< T >::multi_iterator_permutation (
            T B,
            T N,
            size_t k)  [inline], [explicit]
```

Construct a multi_iterator with upper limit N and size k .

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), and GiNaC::basic_multi_iterator< T >::N.

8.101.2.3 multi_iterator_permutation() [3/3]

```
template<class T>
GiNaC::multi_iterator_permutation< T >::multi_iterator_permutation (
            T B,
            T N,
            const std::vector< T > & vv)  [inline], [explicit]
```

Construct from a vector.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::N, and GiNaC::basic_multi_iterator< T >::v.

8.101.3 Member Function Documentation

8.101.3.1 init()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_permutation< T >::init (
            void )  [inline], [virtual]
```

Initialize the multi-index to.

$$(n_1, n_2, n_3, ..., n_k) = (B+0, B+1, B+2, ..., B+k-1)$$

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, and GiNaC::basic_multi_iterator< T >::v.

8.101.3.2 operator++()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_permutation< T >::operator++ (
            int )  [inline], [virtual]
```

The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

If n is in the last configuration and the increment operator ++ is applied to n, the overflow flag will be raised.

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, GiNaC::basic_multi_iterator< T >::N, GiNaC::basic_multi_iterator< T >::size(), and GiNaC::basic_multi_iterator< T >::v.

8.101.3.3 get_sign()

```
template<class T>
int GiNaC::multi_iterator_permutation< T >::get_sign (
            void ) const  [inline]
```

Returns the sign of the permutation, defined by.

$$(-1)^{n_{inv}},$$

where n_{inv} is the number of inversions, e.g. the number of pairs $i < j$ for which

$$n_i > n_j.$$

References GiNaC::basic_multi_iterator< T >::size(), and GiNaC::basic_multi_iterator< T >::v.

8.101.4 Friends And Related Symbol Documentation

8.101.4.1 operator<<

```
template<class T>
template<class TT>
std::ostream & operator<< (
            std::ostream & os,
            const multi_iterator_permutation< TT > & v)  [friend]
```

References multi_iterator_permutation(), and GiNaC::basic_multi_iterator< T >::v.

The documentation for this class was generated from the following file:

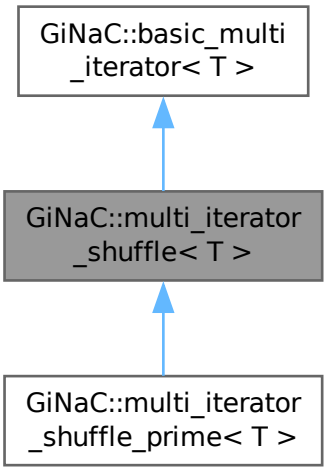
- utils_multi_iterator.h

8.102 GiNaC::multi_iterator_shuffle< T > Class Template Reference

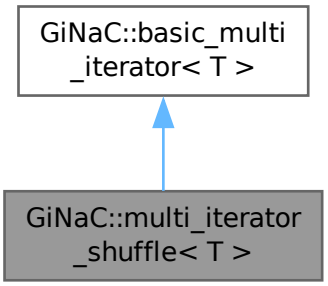
The class multi_iterator_shuffle defines a multi_iterator, which runs over all shuffles of a and b.

`#include <utils_multi_iterator.h>`

Inheritance diagram for GiNaC::multi_iterator_shuffle< T >:



Collaboration diagram for GiNaC::multi_iterator_shuffle< T >:



Public Member Functions

- multi_iterator_shuffle (void)
  Default constructor.
- multi_iterator_shuffle (const std::vector< T > &a, const std::vector< T > &b)
  Construct from a vector.
- basic_multi_iterator< T > & init (void)
  Initialize the multi-index to the first shuffle.
- basic_multi_iterator< T > & operator++ (int)
  The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

Public Member Functions inherited from GiNaC::basic_multi_iterator< T >

- basic_multi_iterator (void)
  Default constructor.
- basic_multi_iterator (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N, lower limit B and size k .
- basic_multi_iterator (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- virtual ~basic_multi_iterator ()
  Destructor.
- size_t size (void) const
  Returns the size of a multi_iterator.
- bool overflow (void) const
  Return the overflow flag.
- const std::vector< T > & get_vector (void) const
  Returns a reference to the vector v.
- T operator[ ] (size_t i) const
  Subscription via [].
- T & operator[ ] (size_t i)
  Subscription via [].
- T operator() (size_t i) const
  Subscription via ().
- T & operator() (size_t i)
  Subscription via ().

Protected Attributes

- size_t N_internal
- std::vector< size_t > v_internal
- std::vector< T > v_orig

Protected Attributes inherited from GiNaC::basic_multi_iterator< T >

- T N
- T B
- std::vector< T > v
- bool flag_overflow

Friends

- template<class TT>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_shuffle< TT > &v)

8.102.1 Detailed Description

template<class T>
class GiNaC::multi_iterator_shuffle< T >

The class multi_iterator_shuffle defines a multi_iterator, which runs over all shuffles of a and b.

8.102.2 Constructor & Destructor Documentation

8.102.2.1 multi_iterator_shuffle() [1/2]

```
template<class T>
GiNaC::multi_iterator_shuffle< T >::multi_iterator_shuffle (
            void )  [inline]
```

Default constructor.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), N_internal, v_internal, and v_orig.

Referenced by GiNaC::multi_iterator_shuffle_prime< T >::multi_iterator_shuffle_prime(), GiNaC::multi_iterator_shuffle_prime< T >:: and operator<<.

8.102.2.2 multi_iterator_shuffle() [2/2]

```
template<class T>
GiNaC::multi_iterator_shuffle< T >::multi_iterator_shuffle (
            const std::vector< T > & a,
            const std::vector< T > & b)  [inline], [explicit]
```

Construct from a vector.

References GiNaC::basic_multi_iterator< T >::B, GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), N_internal, GiNaC::basic_multi_iterator< T >::v, v_internal, and v_orig.

8.102.3 Member Function Documentation

8.102.3.1 init()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_shuffle< T >::init (
            void )  [inline], [virtual]
```

Initialize the multi-index to the first shuffle.

Reimplemented from GiNaC::basic_multi_iterator< T >.

Reimplemented in GiNaC::multi_iterator_shuffle_prime< T >.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, GiNaC::basic_multi_iterator< T >::v, v_internal, and v_orig.

Referenced by GiNaC::Gt::regularise().

8.102.3.2 operator++()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_shuffle< T >::operator++ (
            int )  [inline], [virtual]
```

The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

If n is in the last configuration and the increment operator ++ is applied to n, the overflow flag will be raised.

Reimplemented from GiNaC::basic_multi_iterator< T >.

References GiNaC::basic_multi_iterator< T >::basic_multi_iterator(), GiNaC::basic_multi_iterator< T >::flag_overflow, N_internal, GiNaC::basic_multi_iterator< T >::v, v_internal, and v_orig.

8.102.4 Friends And Related Symbol Documentation

8.102.4.1 operator<<

```
template<class T>
template<class TT>
std::ostream & operator<< (
            std::ostream & os,
            const multi_iterator_shuffle< TT > & v)  [friend]
```

References multi_iterator_shuffle(), and GiNaC::basic_multi_iterator< T >::v.

8.102.5 Member Data Documentation

8.102.5.1 N_internal

```
template<class T>
size_t GiNaC::multi_iterator_shuffle< T >::N_internal  [protected]
```

Referenced by multi_iterator_shuffle(), multi_iterator_shuffle(), and operator++().

8.102.5.2 v_internal

```
template<class T>
std::vector<size_t> GiNaC::multi_iterator_shuffle< T >::v_internal  [protected]
```

Referenced by init(), GiNaC::multi_iterator_shuffle_prime< T >::init(), multi_iterator_shuffle(), multi_iterator_shuffle(), and operator++().

8.102.5.3 v_orig

```
template<class T>
std::vector<T> GiNaC::multi_iterator_shuffle< T >::v_orig  [protected]
```

Referenced by init(), GiNaC::multi_iterator_shuffle_prime< T >::init(), multi_iterator_shuffle(), multi_iterator_shuffle(), and operator++().

The documentation for this class was generated from the following file:

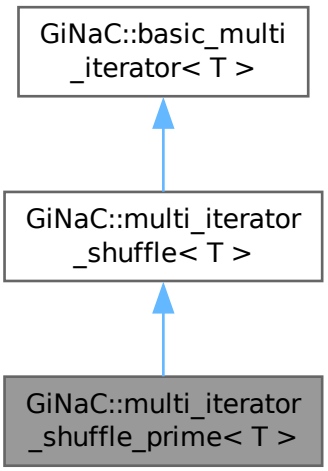
- utils_multi_iterator.h

8.103 GiNaC::multi_iterator_shuffle_prime< T > Class Template Reference

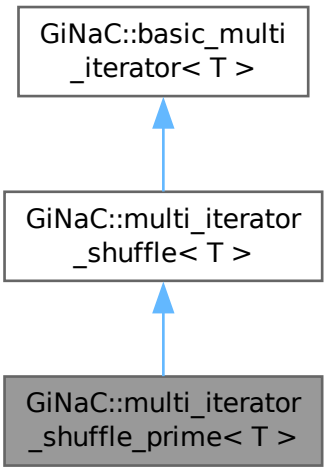
The class multi_iterator_shuffle_prime defines a multi_iterator, which runs over all shuffles of a and b, excluding the first one (a,b).

```
#include <utils_multi_iterator.h>
```

Inheritance diagram for GiNaC::multi_iterator_shuffle_prime< T >:



Collaboration diagram for GiNaC::multi_iterator_shuffle_prime< T >:



Public Member Functions

- multi_iterator_shuffle_prime (void)
  Default constructor.
- multi_iterator_shuffle_prime (const std::vector< T > &a, const std::vector< T > &b)
  Construct from a vector.
- basic_multi_iterator< T > & init (void)
  Initialize the multi-index to the first shuffle.

Public Member Functions inherited from GiNaC::multi_iterator_shuffle< T >

- multi_iterator_shuffle (void)
  Default constructor.
- multi_iterator_shuffle (const std::vector< T > &a, const std::vector< T > &b)
  Construct from a vector.
- basic_multi_iterator< T > & operator++ (int)
  The postfix increment operator allows to write for a multi-index n++, which will update n to the next configuration.

Public Member Functions inherited from GiNaC::basic_multi_iterator< T >

- basic_multi_iterator (void)
  Default constructor.
- basic_multi_iterator (T B, T N, size_t k)
  Construct a multi_iterator with upper limit N, lower limit B and size k .
- basic_multi_iterator (T B, T N, const std::vector< T > &vv)
  Construct from a vector.
- virtual ~basic_multi_iterator ()

Destructor.

- size_t size (void) const

  Returns the size of a multi_iterator.
- bool overflow (void) const

  Return the overflow flag.
- const std::vector< T > & get_vector (void) const

  Returns a reference to the vector v.
- T operator[ ] (size_t i) const

  Subscription via [].
- T & operator[ ] (size_t i)

  Subscription via [].
- T operator() (size_t i) const

  Subscription via ().
- T & operator() (size_t i)

  Subscription via ().

Friends

- template<class TT>
  std::ostream & operator<< (std::ostream &os, const multi_iterator_shuffle_prime< TT > &v)

Additional Inherited Members

Protected Attributes inherited from GiNaC::multi_iterator_shuffle< T >

- size_t N_internal
- std::vector< size_t > v_internal
- std::vector< T > v_orig

Protected Attributes inherited from GiNaC::basic_multi_iterator< T >

- T N
- T B
- std::vector< T > v
- bool flag_overflow

8.103.1 Detailed Description

template<class T>
class GiNaC::multi_iterator_shuffle_prime< T >

The class multi_iterator_shuffle_prime defines a multi_iterator, which runs over all shuffles of a and b, excluding the first one (a,b).

8.103.2 Constructor & Destructor Documentation

8.103.2.1 multi_iterator_shuffle_prime() [1/2]

```
template<class T>
GiNaC::multi_iterator_shuffle_prime< T >::multi_iterator_shuffle_prime (
            void )  [inline]
```

Default constructor.

References GiNaC::multi_iterator_shuffle< T >::multi_iterator_shuffle().

Referenced by operator<<.

8.103.2.2 multi_iterator_shuffle_prime() [2/2]

```
template<class T>
GiNaC::multi_iterator_shuffle_prime< T >::multi_iterator_shuffle_prime (
            const std::vector< T > & a,
            const std::vector< T > & b)  [inline], [explicit]
```

Construct from a vector.

References GiNaC::multi_iterator_shuffle< T >::multi_iterator_shuffle().

8.103.3 Member Function Documentation

8.103.3.1 init()

```
template<class T>
basic_multi_iterator< T > & GiNaC::multi_iterator_shuffle_prime< T >::init (
            void )  [inline], [virtual]
```

Initialize the multi-index to the first shuffle.

Reimplemented from GiNaC::multi_iterator_shuffle< T >.

References GiNaC::basic_multi_iterator< T >::flag_overflow, GiNaC::basic_multi_iterator< T >::v, GiNaC::multi_iterator_shuffle< T and GiNaC::multi_iterator_shuffle< T >::v_orig.

8.103.4 Friends And Related Symbol Documentation

8.103.4.1 operator<<

```
template<class T>
template<class TT>
std::ostream & operator<< (
            std::ostream & os,
            const multi_iterator_shuffle_prime< TT > & v)  [friend]
```

References multi_iterator_shuffle_prime(), and GiNaC::basic_multi_iterator< T >::v.

The documentation for this class was generated from the following file:

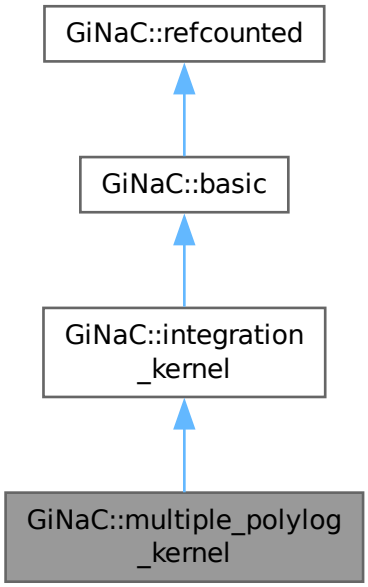
- utils_multi_iterator.h

8.104 GiNaC::multiple_polylog_kernel Class Reference

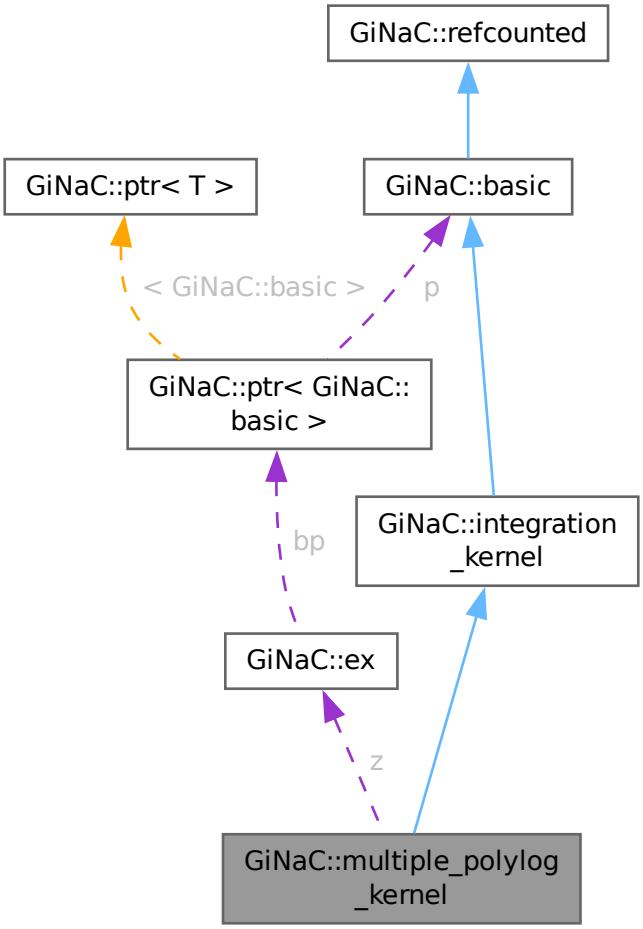
The integration kernel for multiple polylogarithms.

`#include <integration_kernel.h>`

Inheritance diagram for GiNaC::multiple_polylog_kernel:



Collaboration diagram for GiNaC::multiple_polylog_kernel:



Public Member Functions

- multiple_polylog_kernel (const ex &z)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- bool is_numeric (void) const override

  This routine returns true, if the integration kernel can be evaluated numerically.

Public Member Functions inherited from GiNaC::integration_kernel

- ex series (const relational &r, int order, unsigned options=0) const override

Default implementation of ex::series().

- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- virtual ex Laurent_series (const ex &x, int order) const

  Returns the Laurent series, starting possibly with the pole term.
- virtual ex get_numerical_value (const ex &lambda, int N_trunc=0) const

  Evaluates the integrand at lambda.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

Load (deserialize) the object from an archive node.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- cln::cl_N series_coeff_impl (int i) const override

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual bool uses_Laurent_series () const

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex z

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.104.1 Detailed Description

The integration kernel for multiple polylogarithms.

This class represents the differential one-form

$$\omega^{\mathrm{mpl}}(z) = \frac{d\lambda}{\lambda - z}$$

For the case $z = 0$ the class basic_log_kernel should be used.

8.104.2 Constructor & Destructor Documentation

8.104.2.1 multiple_polylog_kernel()

```
GiNaC::multiple_polylog_kernel::multiple_polylog_kernel (
            const ex & z)
```

References multiple_polylog_kernel(), and z.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), and multiple_polylog_kernel().

8.104.3 Member Function Documentation

8.104.3.1 nops()

```
size_t GiNaC::multiple_polylog_kernel::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.104.3.2 op()

```
ex GiNaC::multiple_polylog_kernel::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References z.

8.104.3.3 let_op()

```
ex & GiNaC::multiple_polylog_kernel::let_op (
            size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ensure_if_modifiable(), and z.

8.104.3.4 is_numeric()

```
bool GiNaC::multiple_polylog_kernel::is_numeric (
            void ) const  [override], [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented from GiNaC::integration_kernel.

References GiNaC::ex::evalf(), GiNaC::ex::info(), GiNaC::info_flags::numeric, and z.

8.104.3.5 series_coeff_impl()

```
cln::cl_N GiNaC::multiple_polylog_kernel::series_coeff_impl (
            int i) const  [override], [protected], [virtual]
```

For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.

The i-th coefficient corresponds to the power λ^{i-1}.

Reimplemented from GiNaC::integration_kernel.

References GiNaC::ex::evalf(), GiNaC::ex_to(), and z.

8.104.3.6 do_print()

```
void GiNaC::multiple_polylog_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References GiNaC::ex::print(), GiNaC::print_context::s, and z.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.104.4 Member Data Documentation

8.104.4.1 z

```
ex GiNaC::multiple_polylog_kernel::z  [protected]
```

Referenced by do_print(), is_numeric(), let_op(), multiple_polylog_kernel(), op(), and series_coeff_impl().

The documentation for this class was generated from the following files:

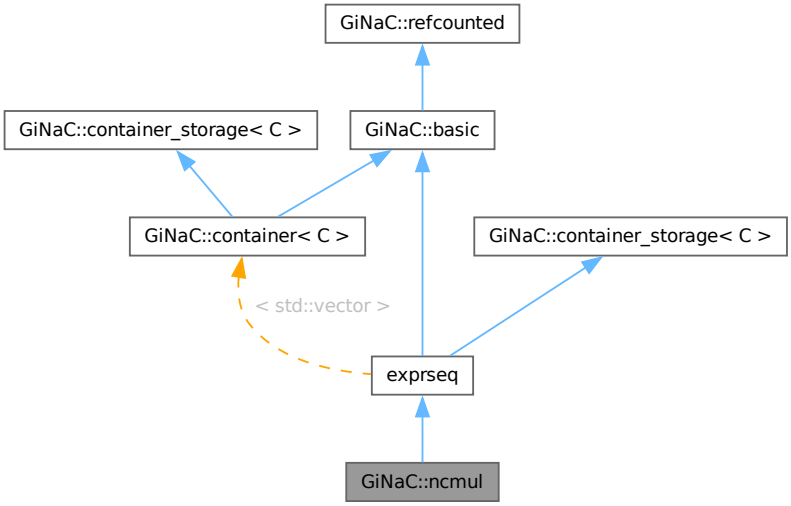
- integration_kernel.h
- integration_kernel.cpp

8.105 GiNaC::ncmul Class Reference

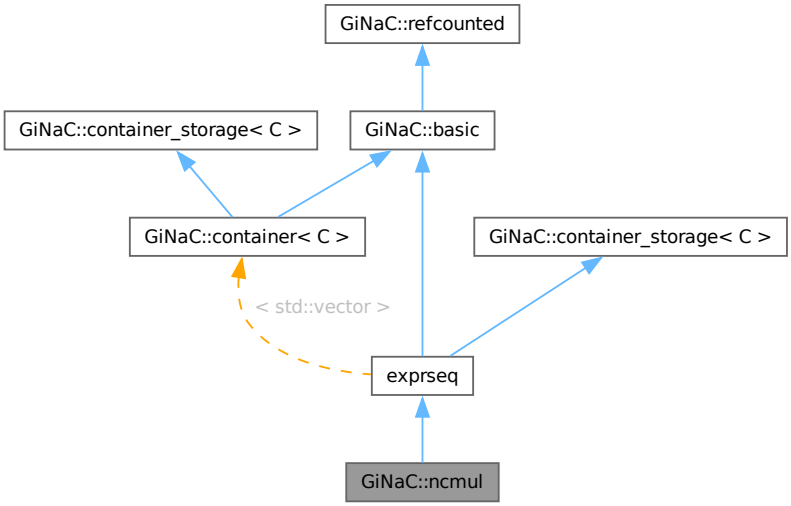
Non-commutative product of expressions.

`#include <ncmul.h>`

Inheritance diagram for GiNaC::ncmul:



Collaboration diagram for GiNaC::ncmul:



Public Member Functions

- ncmul (const ex &lh, const ex &rh)
- ncmul (const ex &f1, const ex &f2, const ex &f3)
- ncmul (const ex &f1, const ex &f2, const ex &f3, const ex &f4)
- ncmul (const ex &f1, const ex &f2, const ex &f3, const ex &f4, const ex &f5)
- ncmul (const ex &f1, const ex &f2, const ex &f3, const ex &f4, const ex &f5, const ex &f6)
- ncmul (const exvector &v)
- ncmul (exvector &&v)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- int degree (const ex &s) const override

  Return degree of highest power in object s.
- int ldegree (const ex &s) const override

  Return degree of lowest power in object s.
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- ex coeff (const ex &s, int n=1) const override

  Return coefficient of degree n in object s.
- ex eval () const override

  Perform automatic term rewriting rules in this class.
- ex evalm () const override

  Evaluate sums, products and integer powers of matrices.
- exvector get_free_indices () const override

  Return a vector containing the free indices of an expression.
- ex thiscontainer (const exvector &v) const override

  Similar to duplicate(), but with a preset sequence.
- ex thiscontainer (exvector &&v) const override

  Similar to duplicate(), but with a preset sequence (which gets pilfered).
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- const exvector & get_factors () const

Public Member Functions inherited from GiNaC::container< std::vector >

- container (STLT const &s)
- container (STLT &&v)
- container (exvector::const_iterator b, exvector::const_iterator e)
- container (std::initializer_list< ex > il)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- void read_archive (const archive_node &n, lst &sym_lst) override

Load (deserialize) the object from an archive node.

- void archive (archive_node &n) const override

  Archive the object.
- container & prepend (const ex &b)

  Add element at front.
- container & append (const ex &b)

  Add element at back.
- container & remove_first ()

  Remove first element.
- container & remove_last ()

  Remove last element.
- container & remove_all ()

  Remove all elements.
- container & sort ()

  Sort elements.
- container & unique ()

  Remove adjacent duplicate elements.
- const_iterator begin () const
- const_iterator end () const
- const_reverse_iterator rbegin () const
- const_reverse_iterator rend () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.

- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for a non-commutative product.
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- void do_print (const print_context &c, unsigned level) const
- void do_print_csrc (const print_context &c, unsigned level) const
- size_t count_factors (const ex &e) const
- void append_factors (exvector &v, const ex &e) const
- exvector expandchildren (unsigned options) const

Protected Member Functions inherited from GiNaC::container< std::vector >

- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- virtual void printseq (const print_context &c, char openbracket, char delim, char closebracket, unsigned this↩ _precedence, unsigned upper_precedence=0) const

  Print sequence of contained elements.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python (const print_python &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const
- STLT subschildren (const exmap &m, unsigned options=0) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Member Functions inherited from GiNaC::container_storage< C >

- container_storage ()
- container_storage (size_t n, const ex &e)
- container_storage (std::initializer_list< ex > il)
- template<class In>
  container_storage (In b, In e)
- void reserve (size_t)
- ~container_storage ()
- void reserve (size_t n)
- void reserve (std::vector< ex > &v, size_t n)

Friends

- class power
- ex reeval_ncmul (const exvector &v)
- ex hold_ncmul (const exvector &v)

Additional Inherited Members

Public Types inherited from GiNaC::container< std::vector >

- typedef STLT::const_iterator const_iterator
- typedef STLT::const_reverse_iterator const_reverse_iterator

Protected Types inherited from GiNaC::container< std::vector >

- typedef container_storage< std::vector >::STLT STLT

Protected Types inherited from GiNaC::container_storage< C >

- typedef C< ex > STLT

Static Protected Member Functions inherited from GiNaC::container< std::vector >

- static unsigned get_default_flags ()
- static char get_open_delim ()
- static char get_close_delim ()

Static Protected Member Functions inherited from GiNaC::container_storage< C >

- static void reserve (STLT &, size_t)

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

Protected Attributes inherited from GiNaC::container_storage< C >

- STLT seq

8.105.1 Detailed Description

Non-commutative product of expressions.

8.105.2 Constructor & Destructor Documentation

8.105.2.1 ncmul() [1/7]

```
GiNaC::ncmul::ncmul (
            const ex & lh,
            const ex & rh)
```

Referenced by GiNaC::canonicalize_clifford(), GiNaC::add::coeff(), GiNaC::color_trace(), GiNaC::cliffordunit::contract_with(), GiNaC::diracgamma::contract_with(), GiNaC::power::eval(), power, and GiNaC::indexed::simplify_indexed_product.

8.105.2.2 ncmul() [2/7]

```
GiNaC::ncmul::ncmul (
            const ex & f1,
            const ex & f2,
            const ex & f3)
```

8.105.2.3 ncmul() [3/7]

```
GiNaC::ncmul::ncmul (
            const ex & f1,
            const ex & f2,
            const ex & f3,
            const ex & f4)
```

8.105.2.4 ncmul() [4/7]

```
GiNaC::ncmul::ncmul (
            const ex & f1,
            const ex & f2,
            const ex & f3,
            const ex & f4,
            const ex & f5)
```

8.105.2.5 ncmul() [5/7]

```
GiNaC::ncmul::ncmul (
            const ex & f1,
            const ex & f2,
            const ex & f3,
            const ex & f4,
            const ex & f5,
            const ex & f6)
```

8.105.2.6 ncmul() [6/7]

```
GiNaC::ncmul::ncmul (
            const exvector & v)
```

8.105.2.7 ncmul() [7/7]

```
GiNaC::ncmul::ncmul (
            exvector && v)
```

8.105.3 Member Function Documentation

8.105.3.1 precedence()

```
unsigned GiNaC::ncmul::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::container< std::vector >.

References precedence().

Referenced by do_print(), do_print_csrc(), and precedence().

8.105.3.2 info()

```
bool GiNaC::ncmul::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::container< std::vector >.

8.105.3.3 degree()

```
int GiNaC::ncmul::degree (
            const ex & s) const  [override], [virtual]
```

Return degree of highest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::ex_to(), GiNaC::basic::is_equal(), and GiNaC::container_storage< C >::seq.

8.105.3.4 ldegree()

```
int GiNaC::ncmul::ldegree (
            const ex & s) const  [override], [virtual]
```

Return degree of lowest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::ex_to(), GiNaC::basic::is_equal(), and GiNaC::container_storage< C >::seq.

8.105.3.5 expand()

```
ex GiNaC::ncmul::expand (
            unsigned options = 0) const  [override], [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::dynallocate(), expandchildren(), GiNaC::status_flags::expanded, GiNaC::is_exactly_a(), GiNaC::container< std::vector >::op(), GiNaC::rename_dummy_indices_uniquely(), GiNaC::container_storage< C >::seq, and GiNaC::basic::setflag().

8.105.3.6 coeff()

```
ex GiNaC::ncmul::coeff (
            const ex & s,
            int n = 1) const  [override], [virtual]
```

Return coefficient of degree n in object s.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::ex::coeff(), GiNaC::dynallocate(), GiNaC::ex_to(), GiNaC::basic::is_equal(), GiNaC::ex::is_zero(), and GiNaC::container_storage< C >::seq.

8.105.3.7 eval()

```
ex GiNaC::ncmul::eval () const  [override], [virtual]
```

Perform automatic term rewriting rules in this class.

In the following x, x1, x2,... stand for a symbolic variables of type ex and c, c1, c2... stand for such expressions that contain a plain number.

- ncmul(...,*(x1,x2),...,ncmul(x3,x4),...) -> ncmul(...,x1,x2,...,x3,x4,...) (associativity)
- ncmul(x) -> x
- ncmul() -> 1
- ncmul(...,c1,...,c2,...) -> *(c1,c2,ncmul(...)) (pull out commutative elements)
- ncmul(x1,y1,x2,y2) -> *(ncmul(x1,x2),ncmul(y1,y2)) (collect elements of same type)
- ncmul(x1,x2,x3,...) -> x::eval_ncmul(x1,x2,x3,...)

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, append_factors(), GiNaC::return_types::commutative, count_factors(), GiNaC::dynallocate(), GiNaC::status_flags::evaluated, GiNaC::factor(), GiNaC::basic::flags, GINAC_ASSERT, GiNaC::make_flat_inserter::handle_factor(), GiNaC::return_types::noncommutative, GiNaC::return_types::noncommutative_composite, and GiNaC::container_storage< C >::seq

8.105.3.8 evalm()

```
ex GiNaC::ncmul::evalm () const  [override], [virtual]
```

Evaluate sums, products and integer powers of matrices.

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate(), GiNaC::ex_to(), GiNaC::is_a(), GiNaC::matrix::mul(), and GiNaC::container_storage< C >::seq.

8.105.3.9 get_free_indices()

```
exvector GiNaC::ncmul::get_free_indices () const  [override], [virtual]
```

Return a vector containing the free indices of an expression.

Reimplemented from GiNaC::basic.

References GiNaC::find_free_and_dummy(), GiNaC::ex::get_free_indices(), GiNaC::container< std::vector >::nops(), and GiNaC::container< std::vector >::op().

8.105.3.10 thiscontainer() [1/2]

```
ex GiNaC::ncmul::thiscontainer (
            const exvector & v) const  [override], [virtual]
```

Similar to duplicate(), but with a preset sequence.

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::dynallocate().

8.105.3.11 thiscontainer() [2/2]

```
ex GiNaC::ncmul::thiscontainer (
            exvector && v) const  [override], [virtual]
```

Similar to duplicate(), but with a preset sequence (which gets pilfered).

Must be overridden by derived classes.

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::dynallocate().

8.105.3.12 conjugate()

```
ex GiNaC::ncmul::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::container< std::vector >::begin(), GiNaC::container< std::vector >::conjugate(), GiNaC::dynallocate(), GiNaC::container< std::vector >::end(), GiNaC::is_clifford_tinfo(), GiNaC::return_types::noncommutative, GiNaC::container< std::vector >::nops(), return_type(), and return_type_tinfo().

8.105.3.13 real_part()

```
ex GiNaC::ncmul::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::basic::real_part().

8.105.3.14 imag_part()

```
ex GiNaC::ncmul::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::container< std::vector >.

References GiNaC::basic::imag_part().

8.105.3.15 derivative()

```
ex GiNaC::ncmul::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for a non-commutative product.

It applies the product rule.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate(), GiNaC::container_storage< C >::seq, and GiNaC::ex::swap().

8.105.3.16 return_type()

```
unsigned GiNaC::ncmul::return_type () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative, GiNaC::container< std::vector >::end(), GINAC_ASSERT, GiNaC::return_types::noncommutative, GiNaC::return_types::noncommutative_composite, and GiNaC::container_storage< C >::seq

Referenced by conjugate().

8.105.3.17 return_type_tinfo()

```
return_type_t GiNaC::ncmul::return_type_tinfo () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::make_return_type_t(), GiNaC::return_types::noncommutative, and GiNaC::container_storage< C >::seq.

Referenced by conjugate().

8.105.3.18 do_print()

```
void GiNaC::ncmul::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References precedence(), and GiNaC::container< std::vector >::printseq().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.105.3.19 do_print_csrc()

```
void GiNaC::ncmul::do_print_csrc (
             const print_context & c,
             unsigned level) const  [protected]
```

References precedence(), GiNaC::container< std::vector >::printseq(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.105.3.20 count_factors()

```
size_t GiNaC::ncmul::count_factors (
             const ex & e) const  [protected]
```

References GiNaC::return_types::commutative, count_factors(), GiNaC::is_exactly_a(), GiNaC::ex::nops(), GiNaC::ex::op(), and GiNaC::ex::return_type().

Referenced by count_factors(), and eval().

8.105.3.21 append_factors()

```
void GiNaC::ncmul::append_factors (
             exvector & v,
             const ex & e) const  [protected]
```

References append_factors(), GiNaC::return_types::commutative, GiNaC::is_exactly_a(), GiNaC::ex::nops(), GiNaC::ex::op(), and GiNaC::ex::return_type().

Referenced by append_factors(), and eval().

8.105.3.22 expandchildren()

```
exvector GiNaC::ncmul::expandchildren (
            unsigned options) const  [protected]
```

References GiNaC::are_ex_trivially_equal(), GiNaC::container< std::vector >::end(), GiNaC::ex::expand(), and GiNaC::container_storage< C >::seq.

Referenced by expand().

8.105.3.23 get_factors()

```
const exvector & GiNaC::ncmul::get_factors () const
```

References GiNaC::container_storage< C >::seq.

8.105.4 Friends And Related Symbol Documentation

8.105.4.1 power

```
friend class power  [friend]
```

References hold_ncmul, ncmul(), power, and reeval_ncmul.

Referenced by power.

8.105.4.2 reeval_ncmul

```
ex reeval_ncmul (
            const exvector & v)  [friend]
```

References GiNaC::dynallocate(), and reeval_ncmul.

Referenced by power, and reeval_ncmul.

8.105.4.3 hold_ncmul

```
ex hold_ncmul (
            const exvector & v)  [friend]
```

References GiNaC::_ex1, GiNaC::dynallocate(), GiNaC::status_flags::evaluated, and hold_ncmul.

Referenced by hold_ncmul, and power.

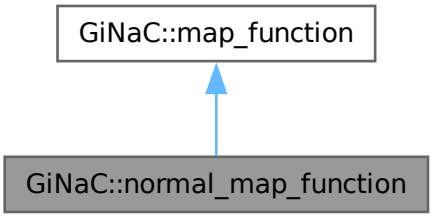
The documentation for this class was generated from the following files:

- ncmul.h
- indexed.cpp
- ncmul.cpp

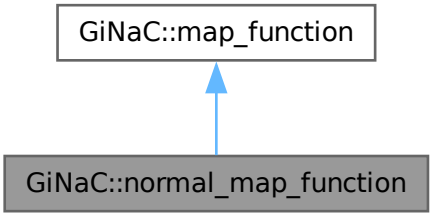
8.106 GiNaC::normal_map_function Struct Reference

Function object to be applied by basic::normal().

Inheritance diagram for GiNaC::normal_map_function:



Collaboration diagram for GiNaC::normal_map_function:



Public Member Functions

- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.106.1 Detailed Description

Function object to be applied by basic::normal().

8.106.2 Member Function Documentation

8.106.2.1 operator()()

```
ex GiNaC::normal_map_function::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References GiNaC::normal().

The documentation for this struct was generated from the following file:

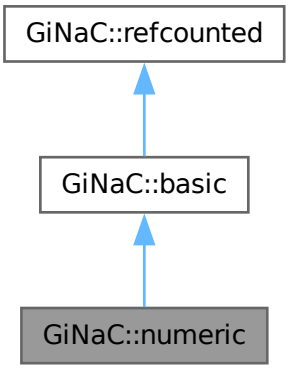
- normal.cpp

8.107 GiNaC::numeric Class Reference

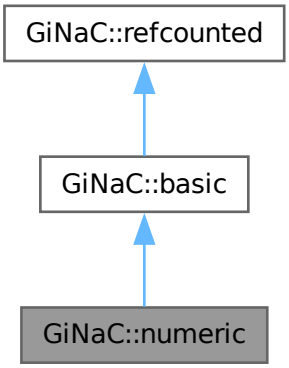
This class is a wrapper around CLN-numbers within the GiNaC class hierarchy.

```
#include <numeric.h>
```

Inheritance diagram for GiNaC::numeric:



Collaboration diagram for GiNaC::numeric:



Public Member Functions

- numeric (int i)
- numeric (unsigned int i)
- numeric (long i)
- numeric (unsigned long i)
- numeric (long long i)
- numeric (unsigned long long i)
- numeric (long numer, long denom)

  Constructor for rational numerics a/b.
- numeric (double d)
- numeric (const char ∗)

  ctor from C-style string.
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- bool is_polynomial (const ex &var) const override

  Check whether this is a polynomial in the given variables.
- int degree (const ex &s) const override

  Return degree of highest power in object s.
- int ldegree (const ex &s) const override

  Return degree of lowest power in object s.
- ex coeff (const ex &s, int n=1) const override

  Return coefficient of degree n in object s.
- bool has (const ex &other, unsigned options=0) const override

  Disassemble real part and imaginary part to scan for the occurrence of a single number.
- ex eval () const override

  Evaluation of numbers doesn't do anything at all.
- ex evalf () const override

  Cast numeric into a floating-point object.

- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const override

  Implementation of ex::normal() for a numeric.
- ex to_rational (exmap &repl) const override

  Implementation of ex::to_rational() for a numeric.
- ex to_polynomial (exmap &repl) const override

  Implementation of ex::to_polynomial() for a numeric.
- numeric integer_content () const override
- ex smod (const numeric &xi) const override

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- numeric max_coefficient () const override

  Implementation ex::max_coefficient().
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- const numeric add (const numeric &other) const

  Numerical addition method.
- const numeric sub (const numeric &other) const

  Numerical subtraction method.
- const numeric mul (const numeric &other) const

  Numerical multiplication method.
- const numeric div (const numeric &other) const

  Numerical division method.
- const numeric power (const numeric &other) const

  Numerical exponentiation.
- const numeric & add_dyn (const numeric &other) const

  Numerical addition method.
- const numeric & sub_dyn (const numeric &other) const

  Numerical subtraction method.
- const numeric & mul_dyn (const numeric &other) const

  Numerical multiplication method.
- const numeric & div_dyn (const numeric &other) const

  Numerical division method.
- const numeric & power_dyn (const numeric &other) const

  Numerical exponentiation.
- const numeric & operator= (int i)
- const numeric & operator= (unsigned int i)
- const numeric & operator= (long i)
- const numeric & operator= (unsigned long i)
- const numeric & operator= (double d)
- const numeric & operator= (const char *s)
- const numeric inverse () const

  Inverse of a number.
- numeric step () const

  Return the step function of a numeric.
- int csgn () const

Return the complex half-plane (left or right) in which the number lies.

- int compare (const numeric &other) const

  This method establishes a canonical order on all numbers.
- bool is_equal (const numeric &other) const
- bool is_zero () const

  True if object is zero.
- bool is_positive () const

  True if object is not complex and greater than zero.
- bool is_negative () const

  True if object is not complex and less than zero.
- bool is_integer () const

  True if object is a non-complex integer.
- bool is_pos_integer () const

  True if object is an exact integer greater than zero.
- bool is_nonneg_integer () const

  True if object is an exact integer greater or equal zero.
- bool is_even () const

  True if object is an exact even integer.
- bool is_odd () const

  True if object is an exact odd integer.
- bool is_prime () const

  Probabilistic primality test.
- bool is_rational () const

  True if object is an exact rational number, may even be complex (denominator may be unity).
- bool is_real () const

  True if object is a real integer, rational or float (but not complex).
- bool is_cinteger () const

  True if object is element of the domain of integers extended by I, i.e.
- bool is_crational () const

  True if object is an exact rational number, may even be complex (denominator may be unity).
- bool operator== (const numeric &other) const
- bool operator!= (const numeric &other) const
- bool operator< (const numeric &other) const

  Numerical comparison: less.
- bool operator<= (const numeric &other) const

  Numerical comparison: less or equal.
- bool operator> (const numeric &other) const

  Numerical comparison: greater.
- bool operator>= (const numeric &other) const

  Numerical comparison: greater or equal.
- int to_int () const

  Converts numeric types to machine's int.
- long to_long () const

  Converts numeric types to machine's long.
- double to_double () const

  Converts numeric types to machine's double.
- cln::cl_N to_cl_N () const

  Returns a new CLN object of type cl_N, representing the value of ∗this.
- const numeric real () const

  Real part of a number.
- const numeric imag () const

Imaginary part of a number.

- const numeric numer () const

  Numerator.

- const numeric denom () const

  Denominator.

- int int_length () const

  Size in binary notation.

- numeric (const cln::cl_N &z)

  Ctor from CLN types.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.

- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.

- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.

- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.

- virtual ex eval_integ () const

  Evaluate integrals, if result is known.

- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.

- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.

- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.

- virtual size_t nops () const

  Number of operands/members.

- virtual ex op (size_t i) const

  Return operand/member at position i.

- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.

- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.

- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

- virtual void accept (GiNaC::visitor &v) const
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.

- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).

- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff for a numeric always returns 0.
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

- void print_numeric (const print_context &c, const char *par_open, const char *par_close, const char *imag↩ _sym, const char *mul_sym, unsigned level) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_csrc (const print_csrc &c, unsigned level) const
- void do_print_csrc_cl_N (const print_csrc_cl_N &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- cln::cl_N value

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.107.1 Detailed Description

This class is a wrapper around CLN-numbers within the GiNaC class hierarchy.

Objects of this type may directly be created by the user.

8.107.2 Constructor & Destructor Documentation

8.107.2.1 numeric() [1/10]

```
GiNaC::numeric::numeric (
            int i)
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), and value.

Referenced by GiNaC::abs(), GiNaC::acos(), GiNaC::acos_eval(), GiNaC::acosh(), GiNaC::acosh_eval(), add(), add_dyn(), GiNaC::pseries::add_series(), GiNaC::asin(), GiNaC::asin_eval(), GiNaC::asinh(), GiNaC::atan(), GiNaC::atan(), GiNaC::atan2_eval(), GiNaC::atanh(), GiNaC::bernoulli(), GiNaC::binomial(), GiNaC::CatalanEvalf(), GiNaC::pseries::coeff(), GiNaC::Eisenstein_h_kernel::coefficient_a0(), GiNaC::Eisenstein_h_kernel::coefficient_an(), compare(), conjugate(), GiNaC::su3d::contract_with(), GiNaC::su3f::contract_with(), GiNaC::su3t::contract_with(), GiNaC::cos(), GiNaC::cosh(), denom(), div(), div_dyn(), GiNaC::doublefactorial(), GiNaC::EllipticE_evalf(), GiNaC::EllipticE_series(), GiNaC::EllipticK_evalf(), GiNaC::EllipticK_series(), GiNaC::EulerEvalf(), GiNaC::power::eval(), evalf(), GiNaC::exp(), GiNaC::power::expand_add(), GiNaC::factorial(), GiNaC::fibonacci(), GiNaC::G2_eval(), GiNaC::G2_evalf(), GiNaC::G3_eval(), GiNaC::G3_evalf(), GiNaC::gcd(), GiNaC::Eisenstein_h_kernel::get_numerical_value(), GiNaC::Eisenstein_kernel::get_numerical_value(), GiNaC::Kronecker_dtau_kernel::get_numerical_value(), GiNaC::modular_form_kernel::get_numerical_value(), GiNaC::H_evalf(), has(), GiNaC::heur_gcd_z(), imag(), imag_part(), integer_content(), inverse(), GiNaC::iquo(), GiNaC::iquo(), GiNaC::irem(), GiNaC::irem(), is_equal(), is_equal_same_type(), GiNaC::modular_form_kernel::is_numeric(), GiNaC::user_defined_kernel::is_numeric(), GiNaC::isqrt(), GiNaC::iterated_integral_evalf_impl(), GiNaC::lcm(), GiNaC::clifford::let_op(), GiNaC::lgamma(), GiNaC::Li2(), GiNaC::Li2_series(), GiNaC::Li_eval(), GiNaC::Li_evalf(), GiNaC::Li_series(), GiNaC::log(), max_coefficient(), GiNaC::mod(), mul(), mul_dyn(), GiNaC::pseries::mul_series(), GiNaC::multinomial_coefficient(), normal(), numer(), operator!=(), operator<(), operator<=(), operator=(), operator=(), operator=(), operator=(), operator=(), operator=(), operator==(), operator>(), operator>=(), GiNaC::Order_series(), GiNaC::PiEvalf(), power(), power_dyn(), GiNaC::Eisenstein_h_kernel::q_expansion_modular_form(), real(), real_part(), GiNaC::S_eval(), GiNaC::S_evalf(), GiNaC::S_series(), GiNaC::symbol::series(), GiNaC::Kronecker_dz_kernel::series_coeff_impl(), GiNaC::sin(), GiNaC::sinh(), smod(), GiNaC::smod(), GiNaC::sqrt(), step(), sub(), sub_dyn(), GiNaC::symm(), GiNaC::tan(), GiNaC::tanh(), GiNaC::tgamma(), to_polynomial(), to_rational(), GiNaC::zeta(), GiNaC::zeta1_evalf(), and GiNaC::zeta2_evalf().

8.107.2.2 numeric() [2/10]

```
GiNaC::numeric::numeric (
            unsigned int i)
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), and value.

8.107.2.3 numeric() [3/10]

```
GiNaC::numeric::numeric (
            long i)
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), and value.

8.107.2.4 numeric() [4/10]

```
GiNaC::numeric::numeric (
            unsigned long i)
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), and value.

8.107.2.5 numeric() [5/10]

```
GiNaC::numeric::numeric (
            long long i)
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), and value.

8.107.2.6 numeric() [6/10]

```
GiNaC::numeric::numeric (
            unsigned long long i)
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), and value.

8.107.2.7 numeric() [7/10]

```
GiNaC::numeric::numeric (
            long numer,
            long denom)
```

Constructor for rational numerics a/b.

Exceptions

overflow_error	(division by zero)

References denom(), GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, numer(), GiNaC::basic::setflag(), and value.

8.107.2.8 numeric() [8/10]

```
GiNaC::numeric::numeric (
            double d)
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), and value.

8.107.2.9 numeric() [9/10]

```
GiNaC::numeric::numeric (
            const char * s)
```

ctor from C-style string.

It also accepts complex numbers in GiNaC notation like "2+5∗I".

References GiNaC::Digits, GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), and value.

8.107.2.10 numeric() [10/10]

```
GiNaC::numeric::numeric (
            const cln::cl_N & z)  [explicit]
```

Ctor from CLN types.

This is for the initiated user or internal use only.

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), and value.

8.107.3 Member Function Documentation

8.107.3.1 precedence()

```
unsigned GiNaC::numeric::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

Referenced by print_numeric().

8.107.3.2 info()

```
bool GiNaC::numeric::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References GiNaC::info_flags::cinteger, GiNaC::info_flags::cinteger_polynomial, GiNaC::info_flags::crational, GiNaC::info_flags::crational_polynomial, GiNaC::info_flags::even, GiNaC::info_flags::expanded, GiNaC::info_flags::integer, GiNaC::info_flags::integer_polynomial, is_cinteger(), is_crational(), is_even(), is_integer(), is_negative(), is_nonneg_integer(), is_odd(), is_pos_integer(), is_positive(), is_prime(), is_rational(), is_real(), is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::negint, GiNaC::info_flags::nonnegative, GiNaC::info_flags::nonnegint, GiNaC::info_flags::numeric, GiNaC::info_flags::odd, GiNaC::info_flags::polynomial, GiNaC::info_flags::posint, GiNaC::info_flags::positive, GiNaC::info_flags::prime, GiNaC::info_flags::rational, GiNaC::info_flags::rational_function, GiNaC::info_flags::rational_polynomial, and GiNaC::info_flags::real.

Referenced by GiNaC::abs_power(), GiNaC::csgn_power(), and GiNaC::zeta1_eval().

8.107.3.3 is_polynomial()

```
bool GiNaC::numeric::is_polynomial (
            const ex & var) const  [override], [virtual]
```

Check whether this is a polynomial in the given variables.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.107.3.4 degree()

```
int GiNaC::numeric::degree (
            const ex & s) const  [override], [virtual]
```

Return degree of highest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.107.3.5 ldegree()

```
int GiNaC::numeric::ldegree (
            const ex & s) const  [override], [virtual]
```

Return degree of lowest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.107.3.6 coeff()

```
ex GiNaC::numeric::coeff (
            const ex & s,
            int n = 1) const  [override], [virtual]
```

Return coefficient of degree n in object s.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, and GiNaC::basic::ex.

Referenced by GiNaC::pseries::mul_const().

8.107.3.7 has()

```
bool GiNaC::numeric::has (
             const ex & other,
             unsigned options = 0) const  [override], [virtual]
```

Disassemble real part and imaginary part to scan for the occurrence of a single number.

Also handles the imaginary unit. It ignores the sign on both this and the argument, which may lead to what might appear as funny results: (2+I).has(-2) -> true. But this is consistent, since we also would like to have (-2+I).has(2) -> true and we want to think about the sign as a multiplicative factor.

Reimplemented from GiNaC::basic.

References GiNaC::_num0_p, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::I, imag(), is_equal(), GiNaC::is_exactly_a(), is_real(), is_zero(), numeric(), and real().

8.107.3.8 eval()

```
ex GiNaC::numeric::eval () const  [override], [virtual]
```

Evaluation of numbers doesn't do anything at all.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, and GiNaC::basic::hold().

8.107.3.9 evalf()

```
ex GiNaC::numeric::evalf () const  [override], [virtual]
```

Cast numeric into a floating-point object.

For example exact numeric(1) is returned as a 1.00000000000000000000 and so on according to how Digits is currently set. In case the object already was a floating point number the precision is trimmed to match the currently set default.

Returns

an ex-handle to a numeric.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, numeric(), and value.

Referenced by GiNaC::Gt::to_numeric().

8.107.3.10 subs()

```
ex GiNaC::numeric::subs (
             const exmap & m,
             unsigned options = 0) const  [inline], [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, and GiNaC::basic::subs_one_level().

8.107.3.11 normal()

```
ex GiNaC::numeric::normal (
             exmap & repl,
             exmap & rev_lookup,
             lst & modifier) const  [override], [virtual]
```

Implementation of ex::normal() for a numeric.

It splits complex numbers into re+I∗im and replaces I and non-rational real numbers with a temporary symbol.

See also

ex::normal

Reimplemented from GiNaC::basic.

References denom(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::I, imag(), is_integer(), is_rational(), is_real(), numer(), numeric(), real(), and GiNaC::replace_with_symbol().

8.107.3.12 to_rational()

```
ex GiNaC::numeric::to_rational (
             exmap & repl) const  [override], [virtual]
```

Implementation of ex::to_rational() for a numeric.

It splits complex numbers into re+I∗im and replaces I and non-rational real numbers with a temporary symbol.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GiNaC::I, imag(), is_rational(), is_real(), numeric(), real(), and GiNaC::replace_with_symbol().

8.107.3.13 to_polynomial()

```
ex GiNaC::numeric::to_polynomial (
            exmap & repl) const  [override], [virtual]
```

Implementation of ex::to_polynomial() for a numeric.

It splits complex numbers into re+I*im and replaces I and non-integer real numbers with a temporary symbol.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GiNaC::I, imag(), is_integer(), is_real(), numeric(), real(), and GiNaC::replace_with_symbol().

8.107.3.14 integer_content()

```
numeric GiNaC::numeric::integer_content () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::abs(), and numeric().

Referenced by GiNaC::mul::eval().

8.107.3.15 smod()

```
ex GiNaC::numeric::smod (
            const numeric & xi) const  [override], [virtual]
```

Apply symmetric modular homomorphism to an expanded multivariate polynomial.

This function is usually used internally by heur_gcd().

Parameters

xi	modulus

Returns

mapped polynomial

See also

heur_gcd

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, numeric(), and GiNaC::smod().

8.107.3.16 max_coefficient()

```
numeric GiNaC::numeric::max_coefficient () const  [override], [virtual]
```

Implementation ex::max_coefficient().

See also

heur_gcd

Reimplemented from GiNaC::basic.

References GiNaC::abs(), and numeric().

8.107.3.17 conjugate()

```
ex GiNaC::numeric::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, is_real(), numeric(), and value.

8.107.3.18 real_part()

```
ex GiNaC::numeric::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, numeric(), and value.

8.107.3.19 imag_part()

```
ex GiNaC::numeric::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, numeric(), and value.

8.107.3.20 archive()

```
void GiNaC::numeric::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_string(), value, and GiNaC::write_real_float().

8.107.3.21 read_archive()

```
void GiNaC::numeric::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::archive_node::find_string(), GiNaC::read_real_float(), GiNaC::basic::setflag(), and value.

8.107.3.22 derivative()

```
ex GiNaC::numeric::derivative (
            const symbol & s) const  [inline], [override], [protected], [virtual]
```

Implementation of ex::diff for a numeric always returns 0.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.107.3.23 is_equal_same_type()

```
bool GiNaC::numeric::is_equal_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if two objects of same type are equal.

Normally needs not be reimplemented as long as it wasn't overwritten by some parent class, since it just calls compare_same_type(). The reason why this function exists is that sometimes it is easier to determine equality than an order relation and then it can be overridden.

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic(), GINAC_ASSERT, is_equal(), GiNaC::is_exactly_a(), and numeric().

8.107.3.24 calchash()

```
unsigned GiNaC::numeric::calchash () const  [override], [protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References GiNaC::golden_ratio_hash(), GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, GiNaC::basic::setflag(), and value.

8.107.3.25 add()

```
const numeric GiNaC::numeric::add (
            const numeric & other) const
```

Numerical addition method.

Adds argument to ∗this and returns result as a numeric object.

References numeric(), and value.

Referenced by GiNaC::multinomial_coefficient(), GiNaC::operator+(), GiNaC::operator++(), GiNaC::operator++(), GiNaC::operator+=(), GiNaC::operator--(), and GiNaC::operator--().

8.107.3.26 sub()

```
const numeric GiNaC::numeric::sub (
            const numeric & other) const
```

Numerical subtraction method.

Subtracts argument from ∗this and returns result as a numeric object.

References numeric(), and value.

Referenced by GiNaC::operator-(), GiNaC::operator-=(), and GiNaC::tgamma_eval().

8.107.3.27 mul()

```
const numeric GiNaC::numeric::mul (
            const numeric & other) const
```

Numerical multiplication method.

Multiplies ∗this and argument and returns result as a numeric object.

References numeric(), and value.

Referenced by GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::mul::expand(), GiNaC::power::expand_add_2(), GiNaC::lcm(), GiNaC::multinomial_coefficient(), GiNaC::operator∗(), GiNaC::operator∗=(), GiNaC::operator-(), and GiNaC::tgamma_eval().

8.107.3.28 div()

```
const numeric GiNaC::numeric::div (
            const numeric & other) const
```

Numerical division method.

Divides ∗this by argument and returns result as a numeric object.

Exceptions

overflow_error	(division by zero)

References numeric(), and value.

Referenced by GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::multinomial_coefficient(), GiNaC::operator/(), GiNaC::operator/=(), GiNaC::basic::series(), and GiNaC::tgamma_eval().

8.107.3.29 power()

```
const numeric GiNaC::numeric::power (
            const numeric & other) const
```

Numerical exponentiation.

Raises ∗this to the power given as argument and returns result as a numeric object.

References GiNaC::_num0_p, GiNaC::_num1_p, numeric(), and value.

Referenced by GiNaC::binomial(), GiNaC::power::eval(), GiNaC::power::expand_add(), and GiNaC::pow().

8.107.3.30 add_dyn()

```
const numeric & GiNaC::numeric::add_dyn (
            const numeric & other) const
```

Numerical addition method.

Adds argument to ∗this and returns result as a numeric object on the heap. Use internally only for direct wrapping into an ex object, where the result would end up on the heap anyways.

References GiNaC::_num0_p, GiNaC::dynallocate(), numeric(), and value.

8.107.3.31 sub_dyn()

```
const numeric & GiNaC::numeric::sub_dyn (
            const numeric & other) const
```

Numerical subtraction method.

Subtracts argument from ∗this and returns result as a numeric object on the heap. Use internally only for direct wrapping into an ex object, where the result would end up on the heap anyways.

References GiNaC::_num0_p, GiNaC::dynallocate(), numeric(), and value.

8.107.3.32 mul_dyn()

```
const numeric & GiNaC::numeric::mul_dyn (
            const numeric & other) const
```

Numerical multiplication method.

Multiplies ∗this and argument and returns result as a numeric object on the heap. Use internally only for direct wrapping into an ex object, where the result would end up on the heap anyways.

References GiNaC::_num1_p, GiNaC::dynallocate(), numeric(), and value.

Referenced by GiNaC::power::expand_add_2().

8.107.3.33 div_dyn()

```
const numeric & GiNaC::numeric::div_dyn (
            const numeric & other) const
```

Numerical division method.

Divides ∗this by argument and returns result as a numeric object on the heap. Use internally only for direct wrapping into an ex object, where the result would end up on the heap anyways.

Exceptions

overflow_error	(division by zero)

References GiNaC::_num1_p, GiNaC::dynallocate(), numeric(), and value.

8.107.3.34 power_dyn()

```
const numeric & GiNaC::numeric::power_dyn (
            const numeric & other) const
```

Numerical exponentiation.

Raises ∗this to the power given as argument and returns result as a numeric object on the heap. Use internally only for direct wrapping into an ex object, where the result would end up on the heap anyways.

References GiNaC::_num0_p, GiNaC::_num1_p, GiNaC::dynallocate(), numeric(), and value.

8.107.3.35 operator=() [1/6]

```
const numeric & GiNaC::numeric::operator= (
            int i)
```

References numeric(), and operator=().

Referenced by operator=(), operator=(), operator=(), operator=(), operator=(), and operator=().

8.107.3.36 operator=() [2/6]

```
const numeric & GiNaC::numeric::operator= (
            unsigned int i)
```

References numeric(), and operator=().

8.107.3.37 operator=() [3/6]

```
const numeric & GiNaC::numeric::operator= (
            long i)
```

References numeric(), and operator=().

8.107.3.38 operator=() [4/6]

```
const numeric & GiNaC::numeric::operator= (
            unsigned long i)
```

References numeric(), and operator=().

8.107.3.39 operator=() [5/6]

```
const numeric & GiNaC::numeric::operator= (
            double d)
```

References numeric(), and operator=().

8.107.3.40 operator=() [6/6]

```
const numeric & GiNaC::numeric::operator= (
            const char * s)
```

References numeric(), and operator=().

8.107.3.41 inverse()

```
const numeric GiNaC::numeric::inverse () const
```

Inverse of a number.

References numeric(), and value.

Referenced by GiNaC::divide_in_z(), GiNaC::power::eval(), GiNaC::heur_gcd_z(), GiNaC::interpolate(), GiNaC::inverse(), and GiNaC::psi1_eval().

8.107.3.42 step()

```
numeric GiNaC::numeric::step () const
```

Return the step function of a numeric.

The imaginary part of it is ignored because the step function is generally considered real but a numeric may develop a small imaginary part due to rounding errors.

References numeric(), and value.

Referenced by GiNaC::step().

8.107.3.43 csgn()

```
int GiNaC::numeric::csgn () const
```

Return the complex half-plane (left or right) in which the number lies.

csgn(x)==0 for x==0, csgn(x)==1 for Re(x)>0 or Re(x)=0 and Im(x)>0, csgn(x)==-1 for Re(x)<0 or Re(x)=0 and Im(x)<0.

See also

numeric::compare(const numeric &other)

References value.

Referenced by GiNaC::csgn().

8.107.3.44 compare()

```
int GiNaC::numeric::compare (
            const numeric & other) const
```

This method establishes a canonical order on all numbers.

For complex numbers this is not possible in a mathematically consistent way but we need to establish some order and it ought to be fast. So we simply define it to be compatible with our method csgn.

Returns

csgn(∗this-other)

See also

numeric::csgn()

References numeric(), and value.

Referenced by GiNaC::binomial(), and GiNaC::power::eval().

8.107.3.45 is_equal()

```
bool GiNaC::numeric::is_equal (
            const numeric & other) const
```

References numeric(), and value.

Referenced by GiNaC::atan(), GiNaC::cos_eval(), GiNaC::doublefactorial(), GiNaC::power::eval(), GiNaC::exp_eval(), has(), is_equal_same_type(), GiNaC::psi2_eval(), GiNaC::sin_eval(), GiNaC::tan_eval(), and GiNaC::zeta1_eval().

8.107.3.46 is_zero()

```
bool GiNaC::numeric::is_zero () const
```

True if object is zero.

References value.

Referenced by GiNaC::asinh_conjugate(), GiNaC::atan(), GiNaC::atan(), GiNaC::atan_conjugate(), GiNaC::clifford_to_lst(), GiNaC::expairseq::construct_from_2_expairseq(), GiNaC::expairseq::construct_from_expairseq_ex(), GiNaC::csgn_eval(), GiNaC::power::eval(), GiNaC::power::expand_mul(), GiNaC::fibonacci(), GiNaC::fsolve(), GiNaC::gcd(), has(), info(), GiNaC::iquo(), GiNaC::iquo(), GiNaC::irem(), GiNaC::irem(), GiNaC::is_zero(), GiNaC::log(), GiNaC::matrix::pivot(), GiNaC::matrix::pow(), GiNaC::pseries::power_const(), GiNaC::step_eval(), and GiNaC::zeta1_eval().

8.107.3.47 is_positive()

```
bool GiNaC::numeric::is_positive () const
```

True if object is not complex and greater than zero.

References value.

Referenced by GiNaC::power::eval(), GiNaC::power::expand_mul(), GiNaC::Gt::findMoebiusTransform(), info(), GiNaC::is_positive(), GiNaC::psi1_eval(), GiNaC::psi2_eval(), GiNaC::Gt::qExpand(), and GiNaC::tgamma_eval().

8.107.3.48 is_negative()

```
bool GiNaC::numeric::is_negative () const
```

True if object is not complex and less than zero.

References value.

Referenced by GiNaC::bernoulli(), GiNaC::beta_eval(), GiNaC::eta_eval(), GiNaC::eta_evalf(), GiNaC::power::eval(), GiNaC::fibonacci(), info(), GiNaC::is_negative(), GiNaC::pseries::power_const(), and GiNaC::Gt::qExpand().

8.107.3.49 is_integer()

```
bool GiNaC::numeric::is_integer () const
```

True if object is a non-complex integer.

References value.

Referenced by GiNaC::bernoulli(), GiNaC::beta_eval(), GiNaC::binomial(), GiNaC::binomial_sym(), GiNaC::Gt::decomposeIntegration GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::power::expand_mul(), GiNaC::fibonacci(), GiNaC::gcd(), info(), GiNaC::iquo(), GiNaC::iquo(), GiNaC::irem(), GiNaC::irem(), GiNaC::is_integer(), GiNaC::isqrt(), GiNaC::lcm(), GiNaC::mod(), normal(), GiNaC::psi1_eval(), GiNaC::psi2_eval(), GiNaC::smod(), GiNaC::tgamma_eval(), to_int(), GiNaC::Gt::to_integer(), to_long(), to_polynomial(), and GiNaC::zeta1_eval().

8.107.3.50 is_pos_integer()

```
bool GiNaC::numeric::is_pos_integer () const
```

True if object is an exact integer greater than zero.

References value.

Referenced by GiNaC::mul::eval(), GiNaC::power::eval(), GiNaC::ifactor(), info(), GiNaC::is_pos_integer(), and GiNaC::pseries::power_const().

8.107.3.51 is_nonneg_integer()

```
bool GiNaC::numeric::is_nonneg_integer () const
```

True if object is an exact integer greater or equal zero.

References value.

Referenced by GiNaC::binomial(), GiNaC::binomial_sym(), GiNaC::doublefactorial(), GiNaC::factorial(), info(), GiNaC::is_nonneg_integer(), print_numeric(), and GiNaC::Gt::to_integer().

8.107.3.52 is_even()

```
bool GiNaC::numeric::is_even () const
```

True if object is an exact even integer.

References value.

Referenced by GiNaC::fibonacci(), info(), GiNaC::is_even(), GiNaC::kronecker_symbol(), and GiNaC::tgamma_eval().

8.107.3.53 is_odd()

```
bool GiNaC::numeric::is_odd () const
```

True if object is an exact odd integer.

References value.

Referenced by info(), GiNaC::is_discriminant_of_quadratic_number_field(), GiNaC::is_odd(), and GiNaC::matrix::pow().

8.107.3.54 is_prime()

```
bool GiNaC::numeric::is_prime () const
```

Probabilistic primality test.

Returns

true if object is exact integer and prime.

References value.

Referenced by info(), and GiNaC::is_prime().

8.107.3.55 is_rational()

```
bool GiNaC::numeric::is_rational () const
```

True if object is an exact rational number, may even be complex (denominator may be unity).

References value.

Referenced by GiNaC::power::eval(), info(), GiNaC::is_rational(), GiNaC::multiply_lcm(), normal(), and to_rational().

8.107.3.56 is_real()

```
bool GiNaC::numeric::is_real () const
```

True if object is a real integer, rational or float (but not complex).

References value.

Referenced by GiNaC::atan(), GiNaC::atan(), GiNaC::beta_eval(), conjugate(), GiNaC::csgn_eval(), denom(), do_print_csrc(), do_print_csrc_cl_N(), GiNaC::eta_eval(), GiNaC::eta_evalf(), GiNaC::power::eval(), GiNaC::fsolve(), has(), info(), is_cinteger(), is_crational(), GiNaC::is_real(), normal(), numer(), operator<(), operator<=(), operator>(), operator>=(), GiNaC::step_eval(), to_double(), to_polynomial(), to_rational(), and GiNaC::zeta().

8.107.3.57 is_cinteger()

```
bool GiNaC::numeric::is_cinteger () const
```

True if object is element of the domain of integers extended by I, i.e.

is of the form a+b∗I, where a and b are integers.

References is_real(), and value.

Referenced by info(), and GiNaC::is_cinteger().

8.107.3.58 is_crational()

```
bool GiNaC::numeric::is_crational () const
```

True if object is an exact rational number, may even be complex (denominator may be unity).

References is_real(), and value.

Referenced by GiNaC::power::eval(), info(), and GiNaC::is_crational().

8.107.3.59 operator==()

```
bool GiNaC::numeric::operator== (
            const numeric & other) const
```

References numeric(), and value.

8.107.3.60 operator"!=()

```
bool GiNaC::numeric::operator!= (
            const numeric & other) const
```

References numeric(), and value.

8.107.3.61 operator<()

```
bool GiNaC::numeric::operator< (
            const numeric & other) const
```

Numerical comparison: less.

Exceptions

invalid_argument	(complex inequality)

References is_real(), numeric(), and value.

8.107.3.62 operator<=()

```
bool GiNaC::numeric::operator<= (
            const numeric & other) const
```

Numerical comparison: less or equal.

Exceptions

invalid_argument	(complex inequality)

References is_real(), numeric(), and value.

8.107.3.63 operator>()

```
bool GiNaC::numeric::operator> (
            const numeric & other) const
```

Numerical comparison: greater.

Exceptions

invalid_argument	(complex inequality)

References is_real(), numeric(), and value.

8.107.3.64 operator>=()

```
bool GiNaC::numeric::operator>= (
            const numeric & other) const
```

Numerical comparison: greater or equal.

Exceptions

invalid_argument	(complex inequality)

References is_real(), numeric(), and value.

8.107.3.65 to_int()

```
int GiNaC::numeric::to_int () const
```

Converts numeric types to machine's int.

You should check with is_integer() if the number is really an integer before calling this method. You may also consider checking the range first.

References GINAC_ASSERT, is_integer(), and value.

Referenced by GiNaC::bernoulli(), GiNaC::Bernoulli_polynomial(), GiNaC::binomial(), GiNaC::binomial_sym(), GiNaC::power::do_print_csrc(), GiNaC::doublefactorial(), GiNaC::power::eval(), GiNaC::power::expand(), GiNaC::power::expand_mul(), GiNaC::factorial(), GiNaC::generalised_Bernoulli_number(), GiNaC::Kronecker_dtau_kernel::series_co and GiNaC::to_int().

8.107.3.66 to_long()

```
long GiNaC::numeric::to_long () const
```

Converts numeric types to machine's long.

You should check with is_integer() if the number is really an integer before calling this method. You may also consider checking the range first.

References GINAC_ASSERT, is_integer(), and value.

Referenced by GiNaC::power::expand(), GiNaC::power::expand_add(), GiNaC::Gt::to_integer(), and GiNaC::to_long().

8.107.3.67 to_double()

```
double GiNaC::numeric::to_double () const
```

Converts numeric types to machine's double.

You should check with is_real() if the number is really not complex before calling this method.

References GINAC_ASSERT, is_real(), and value.

Referenced by GiNaC::Gt::decomposeIntegrationPath(), GiNaC::Gt::qExpand(), and GiNaC::to_double().

8.107.3.68 to_cl_N()

```
cln::cl_N GiNaC::numeric::to_cl_N () const
```

Returns a new CLN object of type cl_N, representing the value of ∗this.

This method may be used when mixing GiNaC and CLN in one project.

References value.

Referenced by GiNaC::abs(), GiNaC::acos(), GiNaC::acosh(), GiNaC::asin(), GiNaC::asinh(), GiNaC::atan(), GiNaC::atan(), GiNaC::atanh(), GiNaC::cos(), GiNaC::cosh(), GiNaC::exp(), GiNaC::fibonacci(), GiNaC::Gt::findMoebiusTransform(), GiNaC::gcd(), GiNaC::iquo(), GiNaC::iquo(), GiNaC::irem(), GiNaC::irem(), GiNaC::isqrt(), GiNaC::lcm(), GiNaC::lgamma(), GiNaC::Li2(), GiNaC::log(), GiNaC::mod(), GiNaC::Kronecker_dtau_kernel::series_coeff_impl(), GiNaC::sin(), GiNaC::sinh(), GiNaC::smod(), GiNaC::sqrt(), GiNaC::tan(), GiNaC::tanh(), GiNaC::tgamma(), and GiNaC::zeta().

8.107.3.69 real()

```
const numeric GiNaC::numeric::real () const
```

Real part of a number.

References numeric(), and value.

Referenced by GiNaC::atan(), GiNaC::csgn_eval(), GiNaC::Gt::decomposeIntegrationPath(), GiNaC::power::eval(), has(), normal(), GiNaC::pseries::power_const(), GiNaC::Gt::qExpand(), GiNaC::real(), GiNaC::step_eval(), to_polynomial(), to_rational(), and GiNaC::Gt::zisToFundamental().

8.107.3.70 imag()

```
const numeric GiNaC::numeric::imag () const
```

Imaginary part of a number.

References numeric(), and value.

Referenced by GiNaC::atan(), GiNaC::csgn_eval(), GiNaC::Gt::decomposeIntegrationPath(), GiNaC::Gt::findMoebiusTransform(), has(), GiNaC::imag(), normal(), GiNaC::Gt::qExpand(), GiNaC::step_eval(), to_polynomial(), to_rational(), and GiNaC::Gt::zisToFundamental().

8.107.3.71 numer()

```
const numeric GiNaC::numeric::numer () const
```

Numerator.

Computes the numerator of rational numbers, rationalized numerator of complex if real and imaginary part are both rational numbers (i.e numer(4/3+5/6∗I) == 8+5∗I), the number carrying the sign in all other cases.

References is_real(), numeric(), and value.

Referenced by GiNaC::power::eval(), GiNaC::frac_cancel(), normal(), GiNaC::numer(), and numeric().

8.107.3.72 denom()

```
const numeric GiNaC::numeric::denom () const
```

Denominator.

Computes the denominator of rational numbers, common integer denominator of complex if real and imaginary part are both rational numbers (i.e denom(4/3+5/6∗I) == 6), one in all other cases.

References GiNaC::_num1_p, is_real(), numeric(), and value.

Referenced by GiNaC::denom(), GiNaC::power::eval(), GiNaC::frac_cancel(), normal(), and numeric().

8.107.3.73 int_length()

```
int GiNaC::numeric::int_length () const
```

Size in binary notation.

For integers, this is the smallest n >= 0 such that -2^n <= x < 2^n. If x > 0, this is the unique n > 0 such that 2^(n-1) <= x < 2^n.

Returns

number of bits (excluding sign) needed to represent that number in two's complement if it is an integer, 0 otherwise.

References value.

Referenced by GiNaC::heur_gcd_z().

8.107.3.74 print_numeric()

```
void GiNaC::numeric::print_numeric (
            const print_context & c,
            const char * par_open,
            const char * par_close,
            const char * imag_sym,
            const char * mul_sym,
            unsigned level) const  [protected]
```

References is_nonneg_integer(), precedence(), GiNaC::print_real_number(), GiNaC::print_context::s, and value.

Referenced by do_print(), do_print_latex(), and do_print_python_repr().

8.107.3.75 do_print()

```
void GiNaC::numeric::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References print_numeric().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.107.3.76 do_print_latex()

```
void GiNaC::numeric::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

References print_numeric().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.107.3.77 do_print_csrc()

```
void GiNaC::numeric::do_print_csrc (
             const print_csrc & c,
             unsigned level) const  [protected]
```

References GiNaC::is_a(), is_real(), GiNaC::print_real_csrc(), GiNaC::print_context::s, and value.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.107.3.78 do_print_csrc_cl_N()

```
void GiNaC::numeric::do_print_csrc_cl_N (
             const print_csrc_cl_N & c,
             unsigned level) const  [protected]
```

References is_real(), GiNaC::print_real_cl_N(), GiNaC::print_context::s, and value.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.107.3.79 do_print_tree()

```
void GiNaC::numeric::do_print_tree (
             const print_tree & c,
             unsigned level) const  [protected]
```

References GiNaC::basic::flags, GiNaC::basic::hashvalue, GiNaC::print_context::s, and value.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.107.3.80 do_print_python_repr()

```
void GiNaC::numeric::do_print_python_repr (
            const print_python_repr & c,
            unsigned level) const  [protected]
```

References print_numeric(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.107.4 Member Data Documentation

8.107.4.1 value

```
cln::cl_N GiNaC::numeric::value  [protected]
```

Referenced by add(), add_dyn(), archive(), calchash(), compare(), conjugate(), csgn(), denom(), div(), div_dyn(), do_print_csrc(), do_print_csrc_cl_N(), do_print_tree(), evalf(), GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(), imag(), imag_part(), int_length(), inverse(), is_cinteger(), is_crational(), is_equal(), is_even(), is_integer(), is_negative(), is_nonneg_integer(), is_odd(), is_pos_integer(), is_positive(), is_prime(), is_rational(), is_real(), is_zero(), mul(), mul_dyn(), numer(), numeric(), numeric(), numeric(), numeric(), numeric(), numeric(), numeric(), numeric(), numeric(), numeric(), operator!=(), operator<(), operator<=(), operator==(), operator>(), operator>=(), power(), power_dyn(), print_numeric(), read_archive(), real(), real_part(), step(), sub(), sub_dyn(), to_cl_N(), to_double(), to_int(), and to_long().

The documentation for this class was generated from the following files:

- numeric.h
- normal.cpp
- numeric.cpp

8.108 GiNaC::op0_is_equal Struct Reference

```
#include <ex.h>
```

Public Member Functions

- bool operator() (const ex &lh, const ex &rh) const

8.108.1 Member Function Documentation

8.108.1.1 operator()()

```
bool GiNaC::op0_is_equal::operator() (
            const ex & lh,
            const ex & rh) const  [inline]
```

References GiNaC::ex::is_equal(), and GiNaC::ex::op().

The documentation for this struct was generated from the following file:

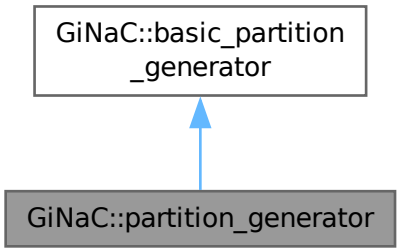
- ex.h

8.109 GiNaC::partition_generator Class Reference

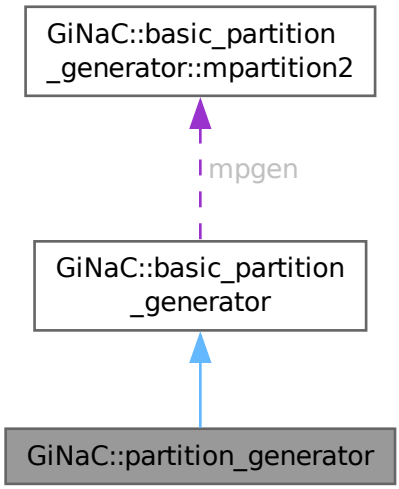
Generate all bounded combinatorial partitions of an integer n with exactly m parts (not including zero parts) in non-decreasing order.

`#include <utils.h>`

Inheritance diagram for GiNaC::partition_generator:



Collaboration diagram for GiNaC::partition_generator:



Public Member Functions

- partition_generator (unsigned n_, unsigned m_)
- const std::vector< unsigned > & get () const
- bool next ()

Private Attributes

- std::vector< unsigned > partition
- bool current_updated

Additional Inherited Members

Protected Member Functions inherited from GiNaC::basic_partition_generator

- basic_partition_generator (unsigned n_, unsigned m_)

Protected Attributes inherited from GiNaC::basic_partition_generator

- mpartition2 mpgen

8.109.1 Detailed Description

Generate all bounded combinatorial partitions of an integer n with exactly m parts (not including zero parts) in non-decreasing order.

8.109.2 Constructor & Destructor Documentation

8.109.2.1 partition_generator()

```
GiNaC::partition_generator::partition_generator (
            unsigned n_,
            unsigned m_)  [inline]
```

References GiNaC::basic_partition_generator::basic_partition_generator(), current_updated, and partition.

8.109.3 Member Function Documentation

8.109.3.1 get()

```
const std::vector< unsigned > & GiNaC::partition_generator::get () const  [inline]
```

References current_updated, GiNaC::basic_partition_generator::mpartition2::m, GiNaC::basic_partition_generator::mpgen, partition, and GiNaC::basic_partition_generator::mpartition2::x.

8.109.3.2 next()

```
bool GiNaC::partition_generator::next ()  [inline]
```

References current_updated, GiNaC::basic_partition_generator::mpgen, and GiNaC::basic_partition_generator::mpartition2::next_pa

8.109.4 Member Data Documentation

8.109.4.1 partition

```
std::vector<unsigned> GiNaC::partition_generator::partition  [mutable], [private]
```

Referenced by get(), and partition_generator().

8.109.4.2 current_updated

```
bool GiNaC::partition_generator::current_updated  [mutable], [private]
```

Referenced by get(), next(), and partition_generator().

The documentation for this class was generated from the following file:

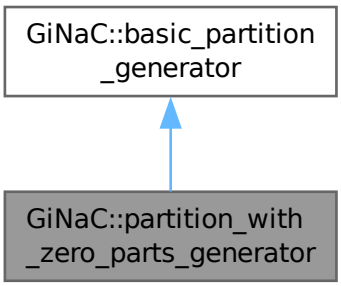
- utils.h

8.110 GiNaC::partition_with_zero_parts_generator Class Reference

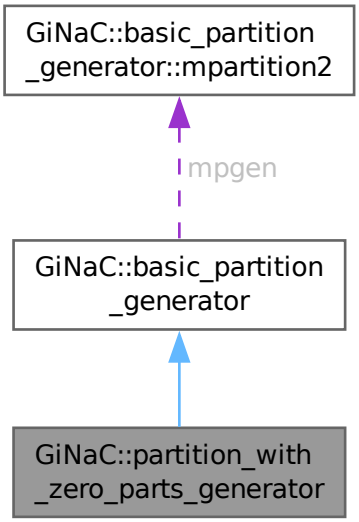
Generate all bounded combinatorial partitions of an integer n with exactly m parts (including zero parts) in non-decreasing order.

```
#include <utils.h>
```

Inheritance diagram for GiNaC::partition_with_zero_parts_generator:



Collaboration diagram for GiNaC::partition_with_zero_parts_generator:



Public Member Functions

- partition_with_zero_parts_generator (unsigned n_, unsigned m_)
- const std::vector< unsigned > & get () const
- bool next ()

Private Attributes

- unsigned m
- std::vector< unsigned > partition
- bool current_updated

Additional Inherited Members

Protected Member Functions inherited from GiNaC::basic_partition_generator

- basic_partition_generator (unsigned n_, unsigned m_)

Protected Attributes inherited from GiNaC::basic_partition_generator

- mpartition2 mpgen

8.110.1 Detailed Description

Generate all bounded combinatorial partitions of an integer n with exactly m parts (including zero parts) in non-decreasing order.

8.110.2 Constructor & Destructor Documentation

8.110.2.1 partition_with_zero_parts_generator()

```
GiNaC::partition_with_zero_parts_generator::partition_with_zero_parts_generator (
            unsigned n_,
            unsigned m_)  [inline]
```

References GiNaC::basic_partition_generator::basic_partition_generator(), current_updated, m, and partition.

8.110.3 Member Function Documentation

8.110.3.1 get()

```
const std::vector< unsigned > & GiNaC::partition_with_zero_parts_generator::get () const
[inline]
```

References current_updated, GiNaC::basic_partition_generator::mpartition2::m, m, GiNaC::basic_partition_generator::mpgen, partition, and GiNaC::basic_partition_generator::mpartition2::x.

Referenced by GiNaC::power::expand_add().

8.110.3.2 next()

```
bool GiNaC::partition_with_zero_parts_generator::next ()  [inline]
```

References current_updated, GiNaC::basic_partition_generator::mpartition2::m, m, GiNaC::basic_partition_generator::mpgen, GiNaC::basic_partition_generator::mpartition2::n, and GiNaC::basic_partition_generator::mpartition2::next_partition().

Referenced by GiNaC::power::expand_add().

8.110.4 Member Data Documentation

8.110.4.1 m

```
unsigned GiNaC::partition_with_zero_parts_generator::m  [private]
```

Referenced by get(), next(), and partition_with_zero_parts_generator().

8.110.4.2 partition

```
std::vector<unsigned> GiNaC::partition_with_zero_parts_generator::partition  [mutable], [private]
```

Referenced by get(), and partition_with_zero_parts_generator().

8.110.4.3 current_updated

```
bool GiNaC::partition_with_zero_parts_generator::current_updated  [mutable], [private]
```

Referenced by get(), next(), and partition_with_zero_parts_generator().

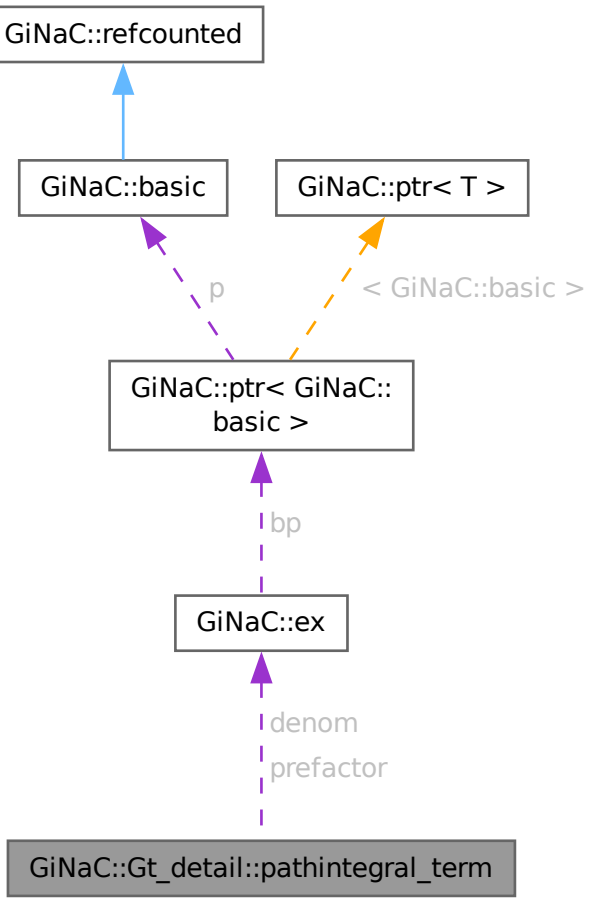
The documentation for this class was generated from the following file:

- utils.h

8.111 GiNaC::Gt_detail::pathintegral_term Class Reference

```
#include <Gt_helpers.h>
```

Collaboration diagram for GiNaC::Gt_detail::pathintegral_term:



Public Member Functions

- pathintegral_term (ex prefactor, int power, const std::vector< ex > &polylog={}, ex denom=0)
- pathintegral_term (const pathintegral_term &a, const pathintegral_term &b, int power_offset=0)
- void integrate (const ex &start, pathintegral_term_list &integrand, pathintegral_term_list &result) const
- ex evaluate (const ex &upper_bound, const std::vector< ex > &path) const

Static Private Member Functions

- static ex G_path (const std::vector< ex > &args, const std::vector< ex > &path)
- static ex G_path (const std::vector< ex > &args, const ex &start, const ex &end)

Private Attributes

- ex prefactor
- int power
- std::vector< ex > polylog
- ex denom

Friends

- class pathintegral_term_list
- std::ostream & operator<< (std::ostream &os, const pathintegral_term &term)
- bool operator< (const pathintegral_term &lh, const pathintegral_term &rh)

8.111.1 Constructor & Destructor Documentation

8.111.1.1 pathintegral_term() [1/2]

```
GiNaC::Gt_detail::pathintegral_term::pathintegral_term (
            ex prefactor,
            int power,
            const std::vector< ex > & polylog = {},
            ex denom = 0)  [inline]
```

References polylog, power, and prefactor.

Referenced by GiNaC::Gt_detail::pathintegral_term_list::add(), GiNaC::Gt_detail::pathintegral_term_list::add(), operator<, operator<<, pathintegral_term(), and pathintegral_term_list.

8.111.1.2 pathintegral_term() [2/2]

```
GiNaC::Gt_detail::pathintegral_term::pathintegral_term (
            const pathintegral_term & a,
            const pathintegral_term & b,
            int power_offset = 0)
```

References denom, GiNaC::ex::is_zero(), GiNaC::is_zero(), pathintegral_term(), polylog, power, and prefactor.

8.111.2 Member Function Documentation

8.111.2.1 integrate()

```
void GiNaC::Gt_detail::pathintegral_term::integrate (
             const ex & start,
             pathintegral_term_list & integrand,
             pathintegral_term_list & result) const
```

References GiNaC::Gt_detail::pathintegral_term_list::add(), GiNaC::class_info< OPT >::first, denom, GiNaC::ex::is_zero(), pathintegral_term_list, polylog, GiNaC::pow(), power, and prefactor.

8.111.2.2 evaluate()

```
ex GiNaC::Gt_detail::pathintegral_term::evaluate (
             const ex & upper_bound,
             const std::vector< ex > & path) const
```

References denom, G_path(), GiNaC::ex::is_zero(), polylog, GiNaC::pow(), power, and prefactor.

8.111.2.3 G_path() [1/2]

```
ex GiNaC::Gt_detail::pathintegral_term::G_path (
             const std::vector< ex > & args,
             const std::vector< ex > & path)  [static], [private]
```

References GiNaC::Gt_detail::deconcatenate_path(), and G_path().

Referenced by evaluate(), and G_path().

8.111.2.4 G_path() [2/2]

```
ex GiNaC::Gt_detail::pathintegral_term::G_path (
             const std::vector< ex > & args,
             const ex & start,
             const ex & end)  [static], [private]
```

References GiNaC::container< C >::append(), GiNaC::factorial(), GiNaC::G(), GiNaC::basic::hold(), and GiNaC::pow().

8.111.3 Friends And Related Symbol Documentation

8.111.3.1 pathintegral_term_list

```
friend class pathintegral_term_list  [friend]
```

References operator<, operator<<, pathintegral_term(), and pathintegral_term_list.

Referenced by integrate(), and pathintegral_term_list.

8.111.3.2 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const pathintegral_term & term)  [friend]
```

References denom, GiNaC::ex::is_zero(), operator<<, pathintegral_term(), polylog, power, and prefactor.

Referenced by operator<<, and pathintegral_term_list.

8.111.3.3 operator<

```
bool operator< (
            const pathintegral_term & lh,
            const pathintegral_term & rh)  [friend]
```

References GiNaC::ex::compare(), denom, GiNaC::ex::is_zero(), operator<, pathintegral_term(), polylog, and power.

Referenced by operator<, and pathintegral_term_list.

8.111.4 Member Data Documentation

8.111.4.1 prefactor

```
ex GiNaC::Gt_detail::pathintegral_term::prefactor  [mutable], [private]
```

Referenced by evaluate(), integrate(), operator<<, pathintegral_term(), and pathintegral_term().

8.111.4.2 power

```
int GiNaC::Gt_detail::pathintegral_term::power  [private]
```

Referenced by evaluate(), integrate(), operator<, operator<<, pathintegral_term(), and pathintegral_term().

8.111.4.3 polylog

```
std::vector<ex> GiNaC::Gt_detail::pathintegral_term::polylog  [private]
```

Referenced by evaluate(), integrate(), operator<, operator<<, pathintegral_term(), and pathintegral_term().

8.111.4.4 denom

```
ex GiNaC::Gt_detail::pathintegral_term::denom  [private]
```

Referenced by evaluate(), integrate(), operator<, operator<<, and pathintegral_term().

The documentation for this class was generated from the following files:

- Gt_helpers.h
- Gt_helpers.cpp

8.112 GiNaC::Gt_detail::pathintegral_term_list Class Reference

`#include <Gt_helpers.h>`

Public Member Functions

- std::set< pathintegral_term >::const_iterator begin () const
- std::set< pathintegral_term >::const_iterator end () const
- void clear ()
- bool empty () const
- size_t size () const
- void add (pathintegral_term &&term)
- template<typename... Args, typename = typename std::enable_if<(sizeof...(Args)>1)>::type>
  void add (Args &&...args)
- void add (const pathintegral_term_list &other)
- pathintegral_term pop ()

Private Attributes

- std::set< pathintegral_term > terms

8.112.1 Member Function Documentation

8.112.1.1 begin()

`std::set< pathintegral_term >::const_iterator GiNaC::Gt_detail::pathintegral_term_list::begin () const [inline]`

References terms.

Referenced by pop().

8.112.1.2 end()

`std::set< pathintegral_term >::const_iterator GiNaC::Gt_detail::pathintegral_term_list::end () const [inline]`

References terms.

Referenced by add().

8.112.1.3 clear()

`void GiNaC::Gt_detail::pathintegral_term_list::clear () [inline]`

References terms.

8.112.1.4 empty()

```
bool GiNaC::Gt_detail::pathintegral_term_list::empty () const  [inline]
```

References terms.

8.112.1.5 size()

```
size_t GiNaC::Gt_detail::pathintegral_term_list::size () const  [inline]
```

References terms.

8.112.1.6 add() [1/3]

```
void GiNaC::Gt_detail::pathintegral_term_list::add (
            pathintegral_term && term)
```

References end(), and terms.

Referenced by GiNaC::Gt_detail::pathintegral_term::integrate().

8.112.1.7 add() [2/3]

```
template<typename...  Args, typename = typename std::enable_if<(sizeof...(Args)>1)>::type>
void GiNaC::Gt_detail::pathintegral_term_list::add (
            Args &&...  args)  [inline]
```

References GiNaC::add::add(), and GiNaC::Gt_detail::pathintegral_term::pathintegral_term().

8.112.1.8 add() [3/3]

```
void GiNaC::Gt_detail::pathintegral_term_list::add (
            const pathintegral_term_list & other)
```

References GiNaC::add::add(), and GiNaC::Gt_detail::pathintegral_term::pathintegral_term().

8.112.1.9 pop()

```
pathintegral_term GiNaC::Gt_detail::pathintegral_term_list::pop ()
```

References begin(), and terms.

8.112.2 Member Data Documentation

8.112.2.1 terms

```
std::set<pathintegral_term> GiNaC::Gt_detail::pathintegral_term_list::terms  [private]
```

Referenced by add(), begin(), clear(), empty(), end(), pop(), and size().

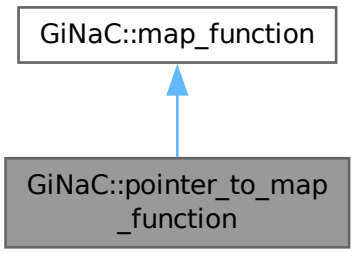
The documentation for this class was generated from the following files:

- Gt_helpers.h
- Gt_helpers.cpp

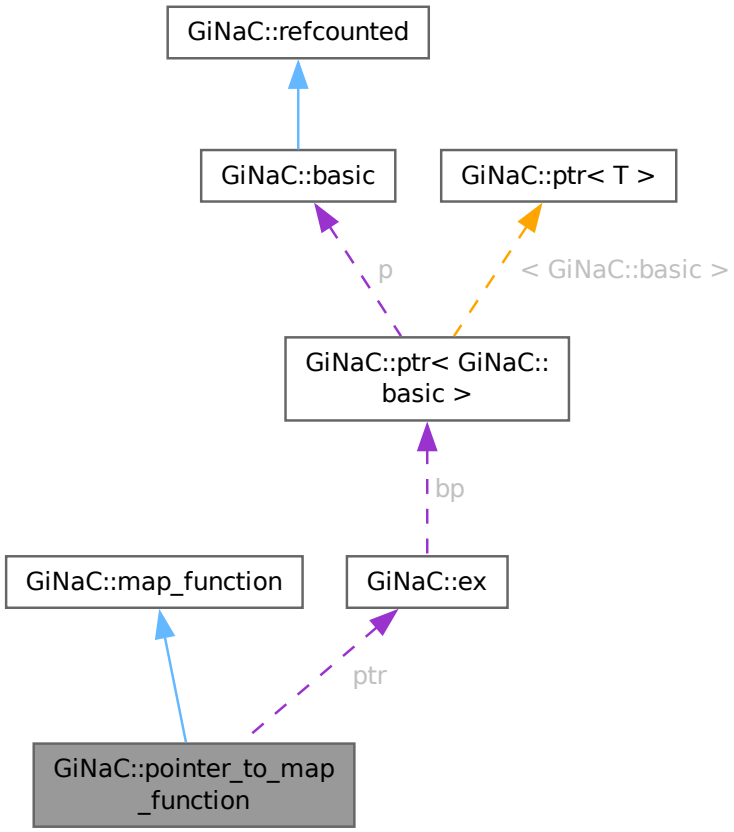
8.113 GiNaC::pointer_to_map_function Class Reference

```
#include <ex.h>
```

Inheritance diagram for GiNaC::pointer_to_map_function:



Collaboration diagram for GiNaC::pointer_to_map_function:



Public Member Functions

- pointer_to_map_function (ex x(const ex &))
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Protected Attributes

- ex(∗) ptr (const ex &)

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.113.1 Constructor & Destructor Documentation

8.113.1.1 pointer_to_map_function()

```
GiNaC::pointer_to_map_function::pointer_to_map_function (
            ex  xconst ex &)  [inline], [explicit]
```

References ptr.

8.113.2 Member Function Documentation

8.113.2.1 operator()()

```
ex GiNaC::pointer_to_map_function::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References GiNaC::ptr< T >::ptr().

8.113.3 Member Data Documentation

8.113.3.1 ptr

```
ex(*) GiNaC::pointer_to_map_function::ptr(const ex &)  [protected]
```

Referenced by pointer_to_map_function().

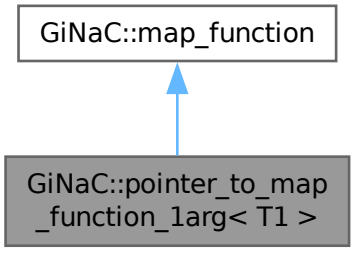
The documentation for this class was generated from the following file:

- ex.h

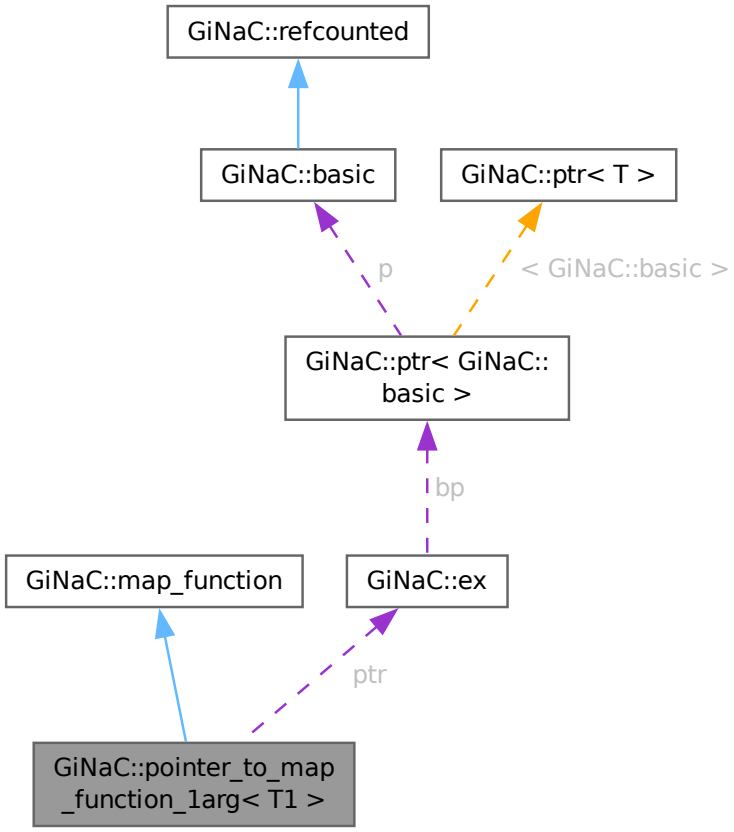
8.114 GiNaC::pointer_to_map_function_1arg< T1 > Class Template Reference

```
#include <ex.h>
```

Inheritance diagram for GiNaC::pointer_to_map_function_1arg< T1 >:



Collaboration diagram for GiNaC::pointer_to_map_function_1arg< T1 >:



Public Member Functions

- pointer_to_map_function_1arg (ex x(const ex &, T1), T1 a1)
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Protected Attributes

- ex(∗) ptr (const ex &, T1)
- T1 arg1

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.114.1 Constructor & Destructor Documentation

8.114.1.1 pointer_to_map_function_1arg()

```
template<class T1>
GiNaC::pointer_to_map_function_1arg< T1 >::pointer_to_map_function_1arg (
            ex  xconst ex &, T1,
            T1 a1)  [inline], [explicit]
```

References arg1, and ptr.

8.114.2 Member Function Documentation

8.114.2.1 operator()()

```
template<class T1>
ex GiNaC::pointer_to_map_function_1arg< T1 >::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References arg1, and GiNaC::ptr< T >::ptr().

8.114.3 Member Data Documentation

8.114.3.1 ptr

```
template<class T1>
ex(*) GiNaC::pointer_to_map_function_1arg< T1 >::ptr(const ex &, T1)  [protected]
```

Referenced by pointer_to_map_function_1arg().

8.114.3.2 arg1

```
template<class T1>
T1 GiNaC::pointer_to_map_function_1arg< T1 >::arg1  [protected]
```

Referenced by operator()(), and pointer_to_map_function_1arg().

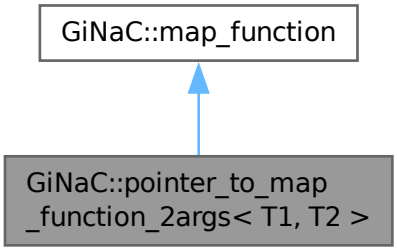
The documentation for this class was generated from the following file:

- ex.h

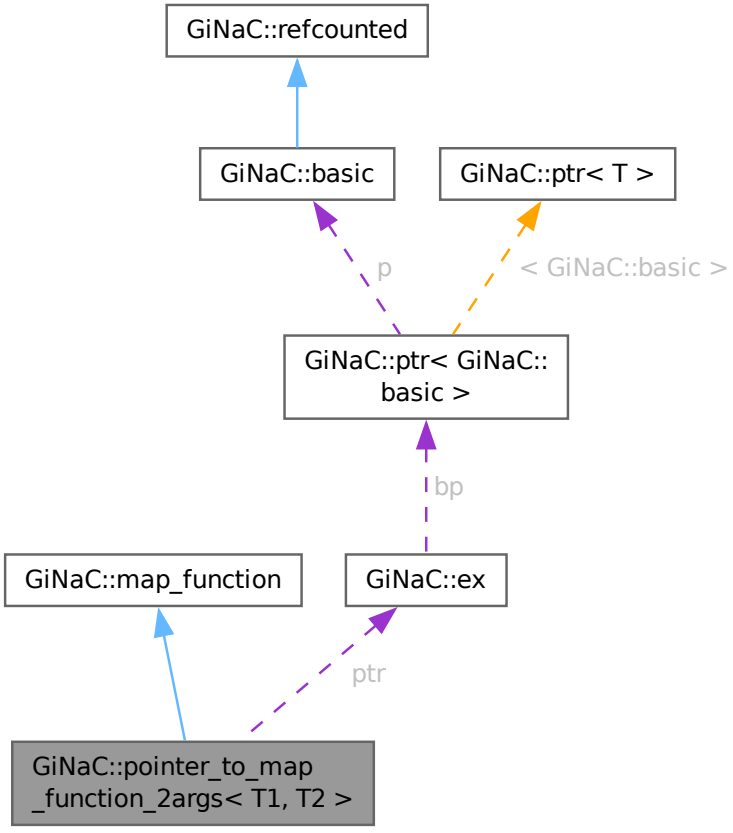
8.115 GiNaC::pointer_to_map_function_2args< T1, T2 > Class Template Reference

`#include <ex.h>`

Inheritance diagram for GiNaC::pointer_to_map_function_2args< T1, T2 >:



Collaboration diagram for GiNaC::pointer_to_map_function_2args< T1, T2 >:



Public Member Functions

- pointer_to_map_function_2args (ex x(const ex &, T1, T2), T1 a1, T2 a2)
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Protected Attributes

- ex(*) ptr (const ex &, T1, T2)
- T1 arg1
- T2 arg2

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.115.1 Constructor & Destructor Documentation

8.115.1.1 pointer_to_map_function_2args()

```
template<class T1, class T2>
GiNaC::pointer_to_map_function_2args< T1, T2 >::pointer_to_map_function_2args (
            ex  xconst ex &, T1, T2,
            T1 a1,
            T2 a2)  [inline], [explicit]
```

References arg1, arg2, and ptr.

8.115.2 Member Function Documentation

8.115.2.1 operator()()

```
template<class T1, class T2>
ex GiNaC::pointer_to_map_function_2args< T1, T2 >::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References arg1, arg2, and GiNaC::ptr< T >::ptr().

8.115.3 Member Data Documentation

8.115.3.1 ptr

```
template<class T1, class T2>
ex(*) GiNaC::pointer_to_map_function_2args< T1, T2 >::ptr(const ex &, T1, T2)  [protected]
```

Referenced by pointer_to_map_function_2args().

8.115.3.2 arg1

```
template<class T1, class T2>
T1 GiNaC::pointer_to_map_function_2args< T1, T2 >::arg1  [protected]
```

Referenced by operator()(), and pointer_to_map_function_2args().

8.115.3.3 arg2

```
template<class T1, class T2>
T2 GiNaC::pointer_to_map_function_2args< T1, T2 >::arg2  [protected]
```

Referenced by operator()(), and pointer_to_map_function_2args().

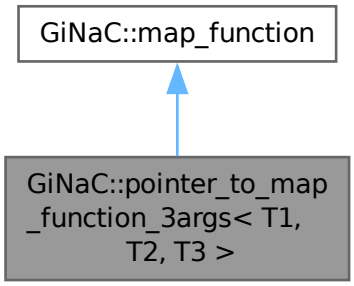
The documentation for this class was generated from the following file:

- ex.h

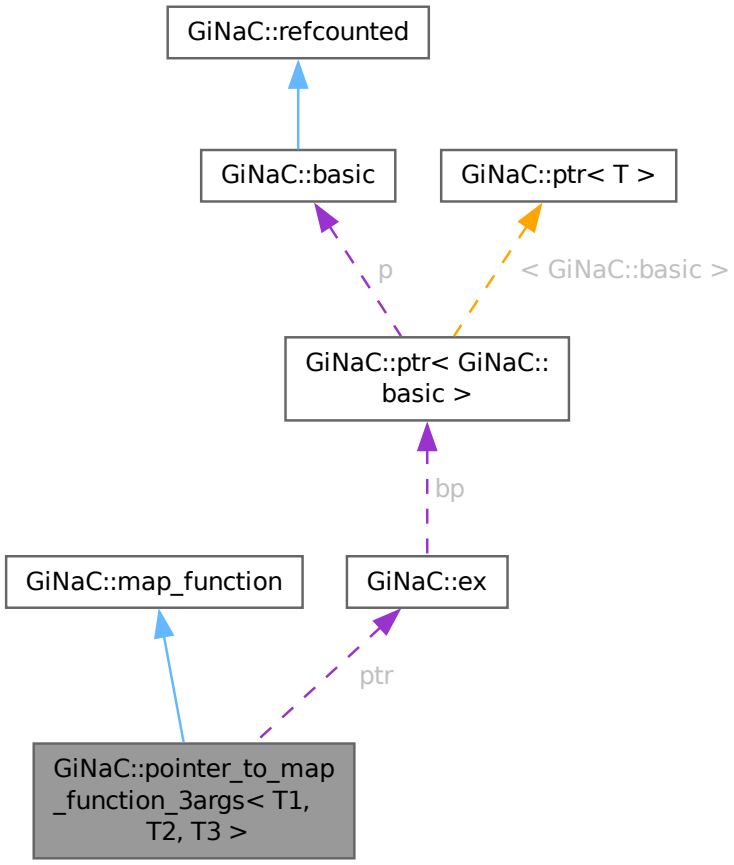
8.116 GiNaC::pointer_to_map_function_3args< T1, T2, T3 > Class Template Reference

```
#include <ex.h>
```

Inheritance diagram for GiNaC::pointer_to_map_function_3args< T1, T2, T3 >:



Collaboration diagram for GiNaC::pointer_to_map_function_3args< T1, T2, T3 >:



Public Member Functions

- pointer_to_map_function_3args (ex x(const ex &, T1, T2, T3), T1 a1, T2 a2, T3 a3)
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ~map_function ()

Protected Attributes

- ex(∗) ptr (const ex &, T1, T2, T3)
- T1 arg1
- T2 arg2
- T3 arg3

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.116.1 Constructor & Destructor Documentation

8.116.1.1 pointer_to_map_function_3args()

```
template<class T1, class T2, class T3>
GiNaC::pointer_to_map_function_3args< T1, T2, T3 >::pointer_to_map_function_3args (
            ex  xconst ex &, T1, T2, T3,
            T1 a1,
            T2 a2,
            T3 a3)  [inline], [explicit]
```

References arg1, arg2, arg3, and ptr.

8.116.2 Member Function Documentation

8.116.2.1 operator()()

```
template<class T1, class T2, class T3>
ex GiNaC::pointer_to_map_function_3args< T1, T2, T3 >::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References arg1, arg2, arg3, and GiNaC::ptr< T >::ptr().

8.116.3 Member Data Documentation

8.116.3.1 ptr

```
template<class T1, class T2, class T3>
ex(*) GiNaC::pointer_to_map_function_3args< T1, T2, T3 >::ptr(const ex &, T1, T2, T3)  [protected]
```

Referenced by pointer_to_map_function_3args().

8.116.3.2 arg1

```
template<class T1, class T2, class T3>
T1 GiNaC::pointer_to_map_function_3args< T1, T2, T3 >::arg1  [protected]
```

Referenced by operator()(), and pointer_to_map_function_3args().

8.116.3.3 arg2

```
template<class T1, class T2, class T3>
T2 GiNaC::pointer_to_map_function_3args< T1, T2, T3 >::arg2  [protected]
```

Referenced by operator()(), and pointer_to_map_function_3args().

8.116.3.4 arg3

```
template<class T1, class T2, class T3>
T3 GiNaC::pointer_to_map_function_3args< T1, T2, T3 >::arg3  [protected]
```

Referenced by operator()(), and pointer_to_map_function_3args().

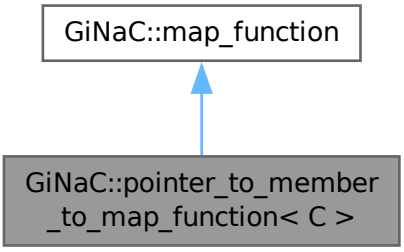
The documentation for this class was generated from the following file:

- ex.h

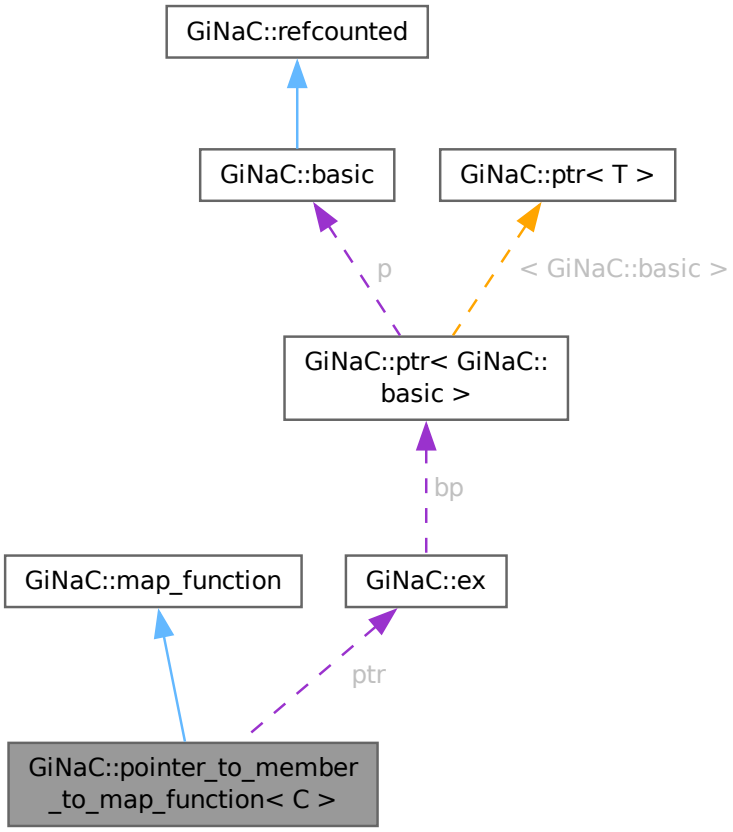
8.117 GiNaC::pointer_to_member_to_map_function< C > Class Template Reference

```
#include <ex.h>
```

Inheritance diagram for GiNaC::pointer_to_member_to_map_function< C >:



Collaboration diagram for GiNaC::pointer_to_member_to_map_function< C >:



Public Member Functions

- pointer_to_member_to_map_function (ex(C::∗member)(const ex &), C &obj)
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ∼map_function ()

Protected Attributes

- ex(C::∗) ptr (const ex &)
- C & c

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.117.1 Constructor & Destructor Documentation

8.117.1.1 pointer_to_member_to_map_function()

```
template<class C>
GiNaC::pointer_to_member_to_map_function< C >::pointer_to_member_to_map_function (
            ex(C::* member )(const ex &),
            C & obj)  [inline], [explicit]
```

References c, and ptr.

8.117.2 Member Function Documentation

8.117.2.1 operator()()

```
template<class C>
ex GiNaC::pointer_to_member_to_map_function< C >::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References c, and ptr.

8.117.3 Member Data Documentation

8.117.3.1 ptr

```
template<class C>
ex(C::*) GiNaC::pointer_to_member_to_map_function< C >::ptr(const ex &)  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function().

8.117.3.2 c

```
template<class C>
C& GiNaC::pointer_to_member_to_map_function< C >::c  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function().

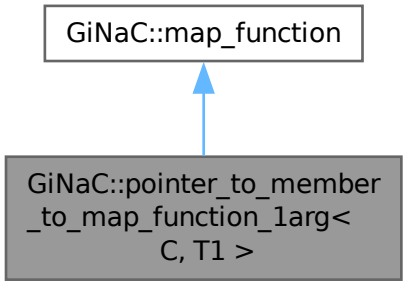
The documentation for this class was generated from the following file:

- ex.h

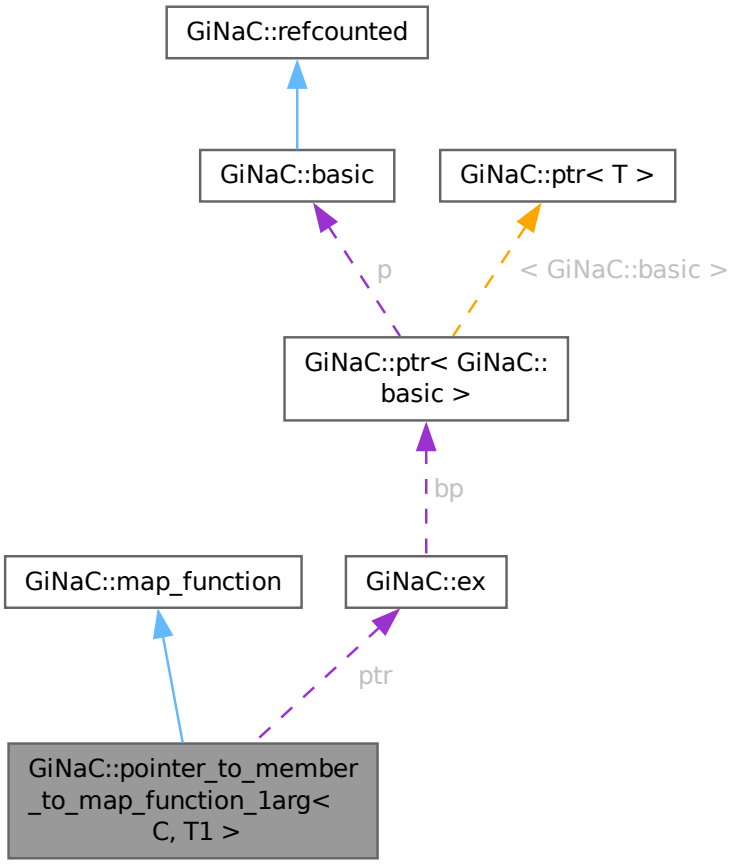
8.118 GiNaC::pointer_to_member_to_map_function_1arg< C, T1 > Class Template Reference

`#include <ex.h>`

Inheritance diagram for GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >:



Collaboration diagram for GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >:



Public Member Functions

- pointer_to_member_to_map_function_1arg (ex(C::∗member)(const ex &, T1), C &obj, T1 a1)
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ∼map_function ()

Protected Attributes

- ex(C::∗) ptr (const ex &, T1)
- C & c
- T1 arg1

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.118.1 Constructor & Destructor Documentation

8.118.1.1 pointer_to_member_to_map_function_1arg()

```
template<class C, class T1>
GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >::pointer_to_member_to_map_function_1arg
(
            ex(C::* member )(const ex &, T1),
            C & obj,
            T1 a1)  [inline], [explicit]
```

References arg1, c, and ptr.

8.118.2 Member Function Documentation

8.118.2.1 operator()()

```
template<class C, class T1>
ex GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References arg1, c, and ptr.

8.118.3 Member Data Documentation

8.118.3.1 ptr

```
template<class C, class T1>
ex(C::*) GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >::ptr(const ex &, T1)  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_1arg().

8.118.3.2 c

```
template<class C, class T1>
C& GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >::c  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_1arg().

8.118.3.3 arg1

```
template<class C, class T1>
T1 GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >::arg1  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_1arg().

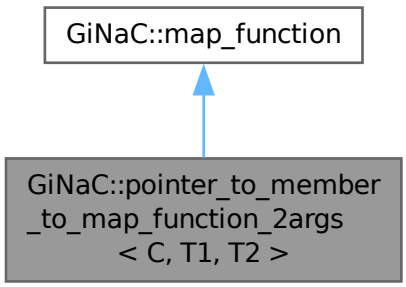
The documentation for this class was generated from the following file:

- ex.h

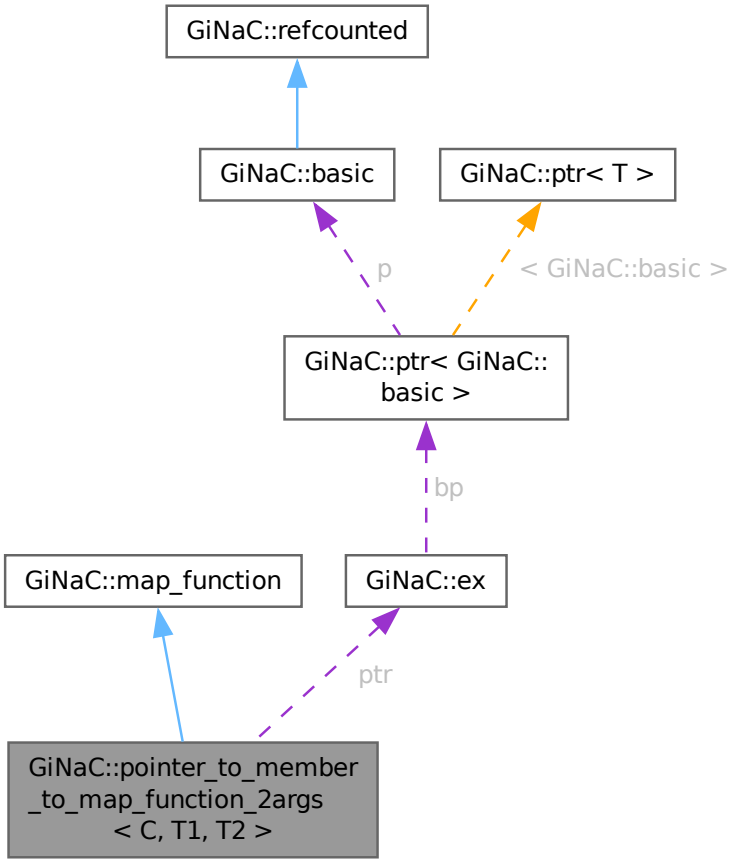
8.119 GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 > Class Template Reference

```
#include <ex.h>
```

Inheritance diagram for GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >:



Collaboration diagram for GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >:



Public Member Functions

- pointer_to_member_to_map_function_2args (ex(C::∗member)(const ex &, T1, T2), C &obj, T1 a1, T2 a2)
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ∼map_function ()

Protected Attributes

- ex(C::∗) ptr (const ex &, T1, T2)
- C & c
- T1 arg1
- T2 arg2

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.119.1 Constructor & Destructor Documentation

8.119.1.1 pointer_to_member_to_map_function_2args()

```
template<class C, class T1, class T2>
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >::pointer_to_member_to_map_function↩
_2args (
            ex(C::* member )(const ex &, T1, T2),
            C & obj,
            T1 a1,
            T2 a2)  [inline], [explicit]
```

References arg1, arg2, c, and ptr.

8.119.2 Member Function Documentation

8.119.2.1 operator()()

```
template<class C, class T1, class T2>
ex GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References arg1, arg2, c, and ptr.

8.119.3 Member Data Documentation

8.119.3.1 ptr

```
template<class C, class T1, class T2>
ex(C::*) GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >::ptr(const ex &, T1, T2)
[protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_2args().

8.119.3.2 c

```
template<class C, class T1, class T2>
C& GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >::c  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_2args().

8.119.3.3 arg1

```
template<class C, class T1, class T2>
T1 GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >::arg1  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_2args().

8.119.3.4 arg2

```
template<class C, class T1, class T2>
T2 GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >::arg2  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_2args().

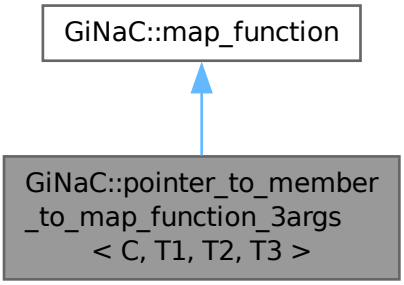
The documentation for this class was generated from the following file:

- ex.h

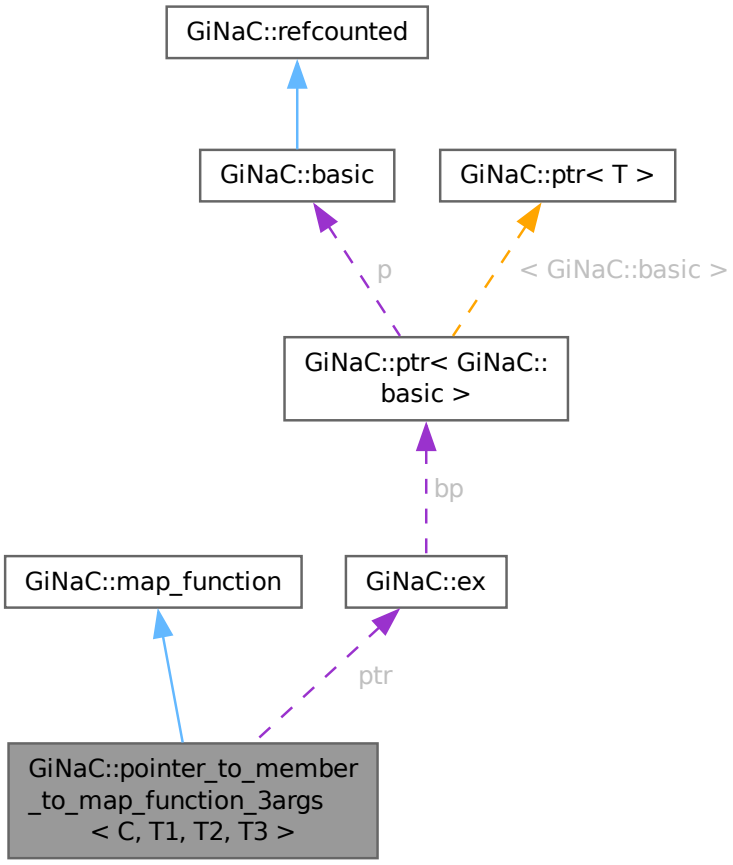
8.120 GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 > Class Template Reference

```
#include <ex.h>
```

Inheritance diagram for GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >:



Collaboration diagram for GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >:



Public Member Functions

- pointer_to_member_to_map_function_3args (ex(C::∗member)(const ex &, T1, T2, T3), C &obj, T1 a1, T2 a2, T3 a3)
- ex operator() (const ex &e) override

Public Member Functions inherited from GiNaC::map_function

- virtual ∼map_function ()

Protected Attributes

- ex(C::∗) ptr (const ex &, T1, T2, T3)
- C & c
- T1 arg1
- T2 arg2
- T3 arg3

Additional Inherited Members

Public Types inherited from GiNaC::map_function

- typedef const ex & argument_type
- typedef ex result_type

8.120.1 Constructor & Destructor Documentation

8.120.1.1 pointer_to_member_to_map_function_3args()

```
template<class C, class T1, class T2, class T3>
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >::pointer_to_member_to_map_↩
function_3args (
            ex(C::* member )(const ex &, T1, T2, T3),
            C & obj,
            T1 a1,
            T2 a2,
            T3 a3)  [inline], [explicit]
```

References arg1, arg2, arg3, c, and ptr.

8.120.2 Member Function Documentation

8.120.2.1 operator()()

```
template<class C, class T1, class T2, class T3>
ex GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >::operator() (
            const ex & e)  [inline], [override], [virtual]
```

Implements GiNaC::map_function.

References arg1, arg2, arg3, c, and ptr.

8.120.3 Member Data Documentation

8.120.3.1 ptr

```
template<class C, class T1, class T2, class T3>
ex(C::*) GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >::ptr(const ex &, T1,
T2, T3)  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_3args().

8.120.3.2 c

```
template<class C, class T1, class T2, class T3>
C& GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >::c  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_3args().

8.120.3.3 arg1

```
template<class C, class T1, class T2, class T3>
T1 GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >::arg1  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_3args().

8.120.3.4 arg2

```
template<class C, class T1, class T2, class T3>
T2 GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >::arg2  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_3args().

8.120.3.5 arg3

```
template<class C, class T1, class T2, class T3>
T3 GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >::arg3  [protected]
```

Referenced by operator()(), and pointer_to_member_to_map_function_3args().

The documentation for this class was generated from the following file:

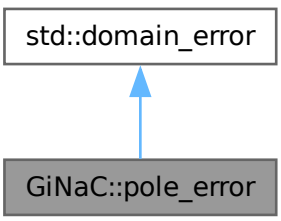
- ex.h

8.121 GiNaC::pole_error Class Reference

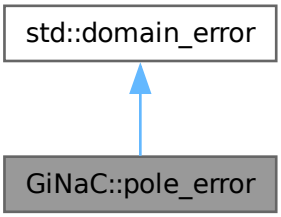
Exception class thrown when a singularity is encountered.

```
#include <numeric.h>
```

Inheritance diagram for GiNaC::pole_error:



Collaboration diagram for GiNaC::pole_error:



Public Member Functions

- pole_error (const std::string &what_arg, int degree)
  ctor for pole_error exception class.
- int degree () const
  Return the degree of the pole_error exception class.

Private Attributes

- int deg

8.121.1 Detailed Description

Exception class thrown when a singularity is encountered.

8.121.2 Constructor & Destructor Documentation

8.121.2.1 pole_error()

```
GiNaC::pole_error::pole_error (
            const std::string & what_arg,
            int degree)  [explicit]
```

ctor for pole_error exception class.

References deg, and degree().

Referenced by GiNaC::atan(), GiNaC::atan(), GiNaC::atan_eval(), GiNaC::atanh_eval(), GiNaC::beta_eval(), GiNaC::power::eval(), GiNaC::lgamma_eval(), GiNaC::log(), GiNaC::log_eval(), GiNaC::pseries::power_const(), GiNaC::psi1_eval(), GiNaC::psi2_eval(), GiNaC::tan_eval(), and GiNaC::tgamma_eval().

8.121.3 Member Function Documentation

8.121.3.1 degree()

```
int GiNaC::pole_error::degree () const
```

Return the degree of the pole_error exception class.

References deg.

Referenced by pole_error().

8.121.4 Member Data Documentation

8.121.4.1 deg

```
int GiNaC::pole_error::deg  [private]
```

Referenced by degree(), and pole_error().

The documentation for this class was generated from the following files:

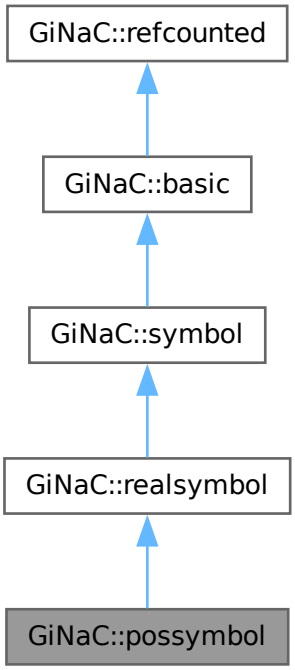
- numeric.h
- utils.cpp

8.122 GiNaC::possymbol Class Reference

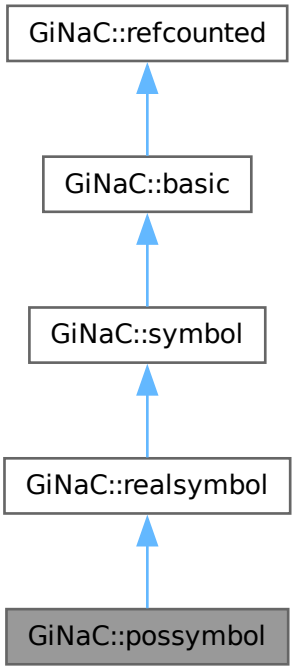
Specialization of symbol to real positive domain.

```
#include <symbol.h>
```

Inheritance diagram for GiNaC::possymbol:



Collaboration diagram for GiNaC::possymbol:



Public Member Functions

- possymbol ()
- possymbol (const std::string &initname)
- possymbol (const std::string &initname, const std::string &texname)
- unsigned get_domain () const override
- possymbol * duplicate () const override

  Create a clone of this object on the heap.

Public Member Functions inherited from GiNaC::realsymbol

- realsymbol ()
- realsymbol (const std::string &initname)
- realsymbol (const std::string &initname, const std::string &texname)
- unsigned get_domain () const override
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- realsymbol * duplicate () const override

  Create a clone of this object on the heap.

Public Member Functions inherited from GiNaC::symbol

- symbol (const std::string &initname)
- symbol (const std::string &initname, const std::string &texname)
- bool info (unsigned inf) const override

  Information about the object.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex evalf () const override

  Evaluate object numerically.
- ex series (const relational &s, int order, unsigned options=0) const override

  Implementation of ex::series() for symbols.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const override

  Implementation of ex::normal() for symbols.
- ex to_rational (exmap &repl) const override

  Implementation of ex::to_rational() for symbols.
- ex to_polynomial (exmap &repl) const override

  Implementation of ex::to_polynomial() for symbols.
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- bool is_polynomial (const ex &var) const override

  Check whether this is a polynomial in the given variables.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- void set_name (const std::string &n)
- void set_TeX_name (const std::string &n)
- std::string get_name () const
- std::string get_TeX_name () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.

- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Additional Inherited Members

Protected Member Functions inherited from GiNaC::symbol

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for single differentiation of a symbol.
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes inherited from GiNaC::symbol

- unsigned serial

  unique serial number for comparison
- std::string name

  printname of this symbol
- std::string TeX_name

  LaTeX name of this symbol.

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.122.1 Detailed Description

Specialization of symbol to real positive domain.

8.122.2 Constructor & Destructor Documentation

8.122.2.1 possymbol() [1/3]

```
GiNaC::possymbol::possymbol ()
```

References GiNaC::realsymbol::realsymbol().

Referenced by duplicate().

8.122.2.2 possymbol() [2/3]

```
GiNaC::possymbol::possymbol (
            const std::string & initname)  [explicit]
```

References GiNaC::realsymbol::realsymbol().

8.122.2.3 possymbol() [3/3]

```
GiNaC::possymbol::possymbol (
            const std::string & initname,
            const std::string & texname)
```

References GiNaC::realsymbol::realsymbol().

8.122.3 Member Function Documentation

8.122.3.1 get_domain()

```
unsigned GiNaC::possymbol::get_domain () const  [inline], [override], [virtual]
```

Reimplemented from GiNaC::symbol.

References GiNaC::domain::positive.

8.122.3.2 duplicate()

```
possymbol * GiNaC::possymbol::duplicate () const  [inline], [override], [virtual]
```

Create a clone of this object on the heap.

One can think of this as simulating a virtual copy constructor which is needed for instance by the refcounted construction of an ex from a basic.

Reimplemented from GiNaC::basic.

References GiNaC::status_flags::dynallocated, possymbol(), and GiNaC::basic::setflag().

The documentation for this class was generated from the following files:

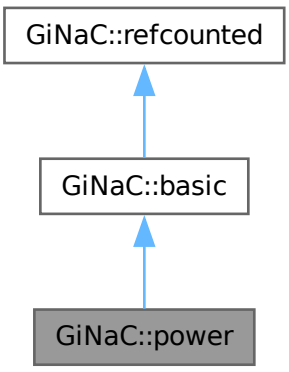
- symbol.h
- symbol.cpp

8.123 GiNaC::power Class Reference

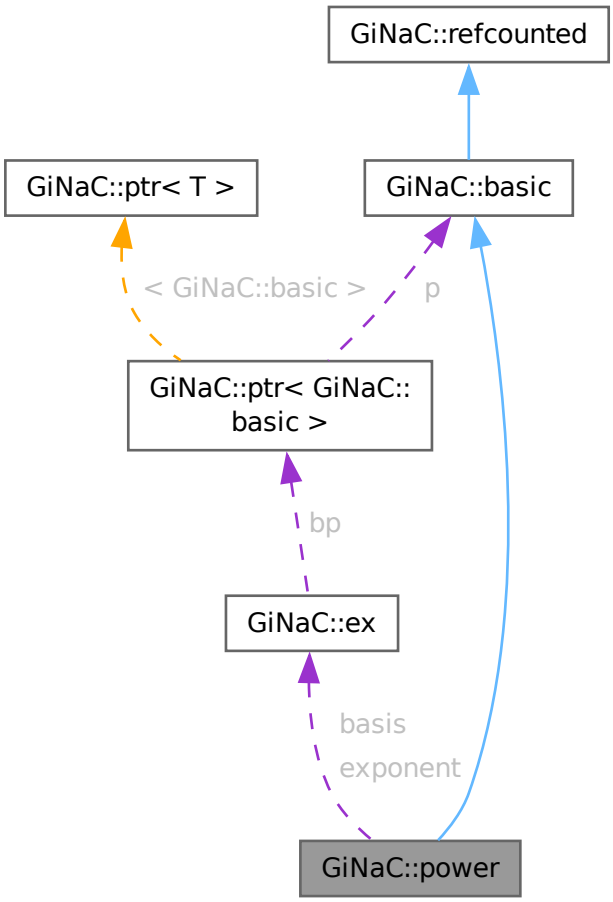
This class holds a two-component object, a basis and and exponent representing exponentiation.

```
#include <power.h>
```

Inheritance diagram for GiNaC::power:



Collaboration diagram for GiNaC::power:



Public Member Functions

- power (const ex &lh, const ex &rh)
- template<typename T>
  power (const ex &lh, const T &rh)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex map (map_function &f) const override

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- bool is_polynomial (const ex &var) const override

  Check whether this is a polynomial in the given variables.

- int degree (const ex &s) const override

  Return degree of highest power in object s.
- int ldegree (const ex &s) const override

  Return degree of lowest power in object s.
- ex coeff (const ex &s, int n=1) const override

  Return coefficient of degree n in object s.
- ex eval () const override

  Perform automatic term rewriting rules in this class.
- ex evalf () const override

  Evaluate object numerically.
- ex evalm () const override

  Evaluate sums, products and integer powers of matrices.
- ex series (const relational &s, int order, unsigned options=0) const override

  Implementation of ex::series() for powers.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- bool has (const ex &other, unsigned options=0) const override

  Test for occurrence of a pattern.
- ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const override

  Implementation of ex::normal() for powers.
- ex to_rational (exmap &repl) const override

  Implementation of ex::to_rational() for powers.
- ex to_polynomial (exmap &repl) const override

  Implementation of ex::to_polynomial() for powers.
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

Little wrapper around printtree to be called within a debugger.

- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual void accept (GiNaC::visitor &v) const
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for a power.
- ex eval_ncmul (const exvector &v) const override
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- void print_power (const print_context &c, const char *powersymbol, const char *openbrace, const char *closebrace, unsigned level) const
- void do_print_dflt (const print_dflt &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_csrc (const print_csrc &c, unsigned level) const
- void do_print_python (const print_python &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const
- void do_print_csrc_cl_N (const print_csrc_cl_N &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Static Protected Member Functions

- static ex expand_add (const add &a, long n, unsigned options)

  expand a^n where a is an add and n is a positive integer.
- static ex expand_add_2 (const add &a, unsigned options)

  Special case of power::expand_add.
- static ex expand_mul (const mul &m, const numeric &n, unsigned options, bool from_expand=false)

  Expand factors of m in m^n where m is a mul and n is an integer.

Protected Attributes

- ex basis
- ex exponent

Protected Attributes inherited from GiNaC::basic

- unsigned flags
  of type status_flags
- unsigned hashvalue
  hash value

Friends

- class mul

8.123.1 Detailed Description

This class holds a two-component object, a basis and and exponent representing exponentiation.

8.123.2 Constructor & Destructor Documentation

8.123.2.1 power() [1/2]

```
GiNaC::power::power (
            const ex & lh,
            const ex & rh)  [inline]
```

References basis, GiNaC::basic::ex, exponent, and power().

Referenced by GiNaC::abs_power(), GiNaC::acos_deriv(), GiNaC::acosh_deriv(), GiNaC::asin_deriv(), GiNaC::asinh_deriv(), GiNaC::atan2_deriv(), GiNaC::atan_deriv(), GiNaC::atanh_deriv(), GiNaC::matrix::charpoly(), GiNaC::mul::combine_overall_coeff(), GiNaC::cos_eval(), GiNaC::cosh_eval(), GiNaC::csgn_power(), GiNaC::mul::do_print_csrc(), do_print_latex(), eval(), GiNaC::integral::eval_integ(), GiNaC::exp_power(), GiNaC::Gt_detail::TransformExpressionWithCache< Type >::impl(), GiNaC::Li2_eval(), GiNaC::Li_eval(), GiNaC::log_deriv(), GiNaC::basic::normal(), GiNaC::operator/(), GiNaC::operator/=(), GiNaC::Order_power(), power(), GiNaC::psi1_series(), GiNaC::psi2_series(), GiNaC::sin_eval(), GiNaC::sinh_eval(), GiNaC::sqrt(), GiNaC::tan_deriv(), GiNaC::tan_eval(), GiNaC::tan_imag_part(), GiNaC::tan_real_part(), GiNaC::tanh_deriv(), GiNaC::tanh_eval(), GiNaC::tanh_imag_part(), and GiNaC::tanh_real_part().

8.123.2.2 power() [2/2]

```
template<typename T>
GiNaC::power::power (
            const ex & lh,
            const T & rh)  [inline]
```

References basis, GiNaC::basic::ex, and exponent.

8.123.3 Member Function Documentation

8.123.3.1 precedence()

```
unsigned GiNaC::power::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

Referenced by print_power().

8.123.3.2 info()

```
bool GiNaC::power::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References basis, GiNaC::info_flags::cinteger_polynomial, GiNaC::basic::clearflag(), GiNaC::info_flags::crational_polynomial, GiNaC::info_flags::even, GiNaC::info_flags::expanded, GiNaC::status_flags::expanded, exponent, GiNaC::basic::flags, GiNaC::info_flags::has_indices, GiNaC::status_flags::has_indices, GiNaC::status_flags::has_no_indices, GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::info_flags::integer_polynomial, GiNaC::info_flags::nonnegative, GiNaC::info_flags::nonnegint, GiNaC::info_flags::polynomial, GiNaC::info_flags::positive, GiNaC::info_flags::rational_function, GiNaC::info_flags::rational_polynomia GiNaC::info_flags::real, and GiNaC::basic::setflag().

8.123.3.3 nops()

```
size_t GiNaC::power::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.123.3.4 op()

```
ex GiNaC::power::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References basis, GiNaC::basic::ex, exponent, and GINAC_ASSERT.

8.123.3.5 map()

```
ex GiNaC::power::map (
            map_function & f) const  [override], [virtual]
```

Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), basis, GiNaC::dynallocate(), GiNaC::basic::ex, and exponent.

8.123.3.6 is_polynomial()

```
bool GiNaC::power::is_polynomial (
            const ex & var) const  [override], [virtual]
```

Check whether this is a polynomial in the given variables.

Reimplemented from GiNaC::basic.

References basis, GiNaC::basic::ex, exponent, GiNaC::ex::has(), GiNaC::ex::info(), GiNaC::ex::is_polynomial(), and GiNaC::info_flags::nonnegint.

8.123.3.7 degree()

```
int GiNaC::power::degree (
            const ex & s) const  [override], [virtual]
```

Return degree of highest power in object s.

Reimplemented from GiNaC::basic.

References basis, GiNaC::ex::degree(), GiNaC::basic::ex, GiNaC::ex_to(), exponent, GiNaC::ex::has(), GiNaC::basic::is_equal(), GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), and GiNaC::is_integer().

8.123.3.8 ldegree()

```
int GiNaC::power::ldegree (
            const ex & s) const  [override], [virtual]
```

Return degree of lowest power in object s.

Reimplemented from GiNaC::basic.

References basis, GiNaC::basic::ex, GiNaC::ex_to(), exponent, GiNaC::ex::has(), GiNaC::basic::is_equal(), GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::is_integer(), and GiNaC::ex::ldegree().

8.123.3.9 coeff()

```
ex GiNaC::power::coeff (
            const ex & s,
            int n = 1) const  [override], [virtual]
```

Return coefficient of degree n in object s.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1, basis, GiNaC::basic::ex, GiNaC::ex_to(), exponent, GiNaC::basic::is_equal(), GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), and GiNaC::is_integer().

Referenced by expand(), and expand_add().

8.123.3.10 eval()

```
ex GiNaC::power::eval () const  [override], [virtual]
```

Perform automatic term rewriting rules in this class.

In the following x, x1, x2,... stand for a symbolic variables of type ex and c, c1, c2... stand for such expressions that contain a plain number.

- ^(x,0) -> 1 (also handles ^(0,0))
- ^(x,1) -> x
- ^(0,c) -> 0 or exception (depending on the real part of c)
- ^(1,x) -> 1
- ^(c1,c2) -> *(c1^n,c1^(c2-n)) (so that 0<(c2-n)<1, try to evaluate roots, possibly in numerator and denominator of c1)
- ^(^(x,c1),c2) -> ^(x,c1*c2) if x is positive and c1 is real.
- ^(^(x,c1),c2) -> ^(x,c1*c2) (c2 integer or -1 < c1 <= 1 or (c1=-1 and c2>0), case c1=1 should not happen, see below!)
- ^(*(x,y,z),c) -> *(x^c,y^c,z^c) (if c integer)
- ^(*(x,c1),c2) -> ^(x,c2)*c1^c2 (c1>0)
- ^(*(x,c1),c2) -> ^(-x,c2)*c1^c2 (c1<0)

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::_ex_1, GiNaC::_num0_p, GiNaC::_num1_p, GiNaC::_num_1_p, GiNaC::abs(), basis, GiNaC::basic::clearflag(), GiNaC::ex::coeff(), GiNaC::return_types::commutative, GiNaC::numeric::compare(), GiNaC::numeric::denom(), GiNaC::numeric::div(), GiNaC::dynallocate(), GiNaC::status_flags::evaluated, GiNaC::basic::ex, GiNaC::ex_to(), expand_mul(), exponent, GiNaC::basic::flags, GINAC_ASSERT, GiNaC::status_flags::hash_calculated, GiNaC::basic::hold(), GiNaC::ex::info(), GiNaC::ex::integer_content(), GiNaC::numeric::inverse(), GiNaC::iquo(), GiNaC::is_a(), GiNaC::numeric::is_crational(), GiNaC::ex::is_equal(), GiNaC::numeric::is_equal(), GiNaC::is_exactly_a(), GiNaC::numeric::is_integer(), GiNaC::is_negative(), GiNaC::numeric::is_negative(), GiNaC::numeric::is_pos_integer(), GiNaC::numeric::is_positive(), GiNaC::numeric::is_rational(), GiNaC::numeric::is_real(), GiNaC::ex::is_zero(), GiNaC::numeric::is_zero(), likely, GiNaC::numeric::mul(), mul, GiNaC::ncmul::ncmul(), GiNaC::numeric::numer(), GiNaC::numeric::numeric(), GiNaC::ex::op(), GiNaC::expairseq::overall_coeff, GiNaC::pole_error::pole_error(), GiNaC::info_flags::positive, GiNaC::pow(), GiNaC::numeric::power(), power(), GiNaC::info_flags::real, GiNaC::numeric::real(), GiNaC::ex::return_type(), GiNaC::expairseq::seq, GiNaC::basic::setflag(), and GiNaC::numeric::to_int().

8.123.3.11 evalf()

```
ex GiNaC::power::evalf () const  [override], [virtual]
```

Evaluate object numerically.

Reimplemented from GiNaC::basic.

References basis, GiNaC::dynallocate(), GiNaC::ex::evalf(), GiNaC::basic::ex, exponent, and GiNaC::is_exactly_a().

8.123.3.12 evalm()

```
ex GiNaC::power::evalm () const  [override], [virtual]
```

Evaluate sums, products and integer powers of matrices.

Reimplemented from GiNaC::basic.

References basis, GiNaC::dynallocate(), GiNaC::ex::evalm(), GiNaC::basic::ex, GiNaC::ex_to(), exponent, GiNaC::is_a(), GiNaC::is_exactly_a(), and GiNaC::pow().

8.123.3.13 series()

```
ex GiNaC::power::series (
             const relational & r,
             int order,
             unsigned options = 0) const  [override], [virtual]
```

Implementation of ex::series() for powers.

This performs Laurent expansion of reciprocals of series at singularities.

See also

ex::series

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, basis, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::exp(), GiNaC::expair::expair(), GiNaC::ex::expand(), exponent, GiNaC::ex::info(), GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::is_integer(), GiNaC::ex::is_zero(), GiNaC::ex::ldegree(), GiNaC::relational::lhs(), GiNaC::log(), GiNaC::info_flags::negint, GiNaC::subs_options::no_pattern, GiNaC::pseries::pseries(), GiNaC::info_flags::rational_function, GiNaC::relational::rhs(), GiNaC::basic::series(), GiNaC::ex::series(), GiNaC::ex::subs(), and GiNaC::to_int().

8.123.3.14 subs()

```
ex GiNaC::power::subs (
             const exmap & m,
             unsigned options = 0) const  [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References GiNaC::subs_options::algebraic, GiNaC::are_ex_trivially_equal(), basis, GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex_to(), exponent, GiNaC::subs_options::no_pattern, GiNaC::pow(), GiNaC::ex::subs(), GiNaC::basic::subs_one_level(), and GiNaC::tryfactsubs().

8.123.3.15 has()

```
bool GiNaC::power::has (
            const ex & pattern,
            unsigned options = 0) const  [override], [virtual]
```

Test for occurrence of a pattern.

An object 'has' a pattern if it matches the pattern itself or one of the children 'has' it. As a consequence (according to the definition of children) given e=x+y+z, e.has(x) is true but e.has(x+y) is false.

Reimplemented from GiNaC::basic.

References GiNaC::has_options::algebraic, basis, GiNaC::basic::ex, GiNaC::ex_to(), exponent, GiNaC::basic::has(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::is_a(), GiNaC::ex::match(), GiNaC::info_flags::negint, GiNaC::ex::op(), and GiNaC::info_flags::posint.

8.123.3.16 normal()

```
ex GiNaC::power::normal (
            exmap & repl,
            exmap & rev_lookup,
            lst & modifier) const  [override], [virtual]
```

Implementation of ex::normal() for powers.

It normalizes the basis, distributes integer exponents to numerator and denominator, and replaces non-integer powers by temporary symbols.

See also

ex::normal

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, basis, GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex_to(), exponent, GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::ex::is_equal(), GiNaC::info_flags::negative, GiNaC::subs_options::no_pattern, GiNaC::container< C >::nops(), GiNaC::container< C >::op(), GiNaC::ex::op(), GiNaC::info_flags::positive, GiNaC::pow(), GiNaC::replace_with_symbol(), and GiNaC::ex::subs().

8.123.3.17 to_rational()

```
ex GiNaC::power::to_rational (
            exmap & repl) const  [override], [virtual]
```

Implementation of ex::to_rational() for powers.

It replaces non-integer powers by temporary symbols.

Reimplemented from GiNaC::basic.

References basis, GiNaC::basic::ex, exponent, GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::pow(), GiNaC::replace_with_symbol(), and GiNaC::ex::to_rational().

8.123.3.18 to_polynomial()

```
ex GiNaC::power::to_polynomial (
            exmap & repl) const  [override], [virtual]
```

Implementation of ex::to_polynomial() for powers.

It replaces non-posint powers by temporary symbols.

Reimplemented from GiNaC::basic.

References GiNaC::_ex_1, basis, GiNaC::collect_common_factors(), GiNaC::basic::ex, exponent, GiNaC::ex::info(), GiNaC::is_exactly_a(), GiNaC::info_flags::negint, GiNaC::info_flags::posint, GiNaC::pow(), GiNaC::replace_with_symbol(), and GiNaC::ex::to_polynomial().

8.123.3.19 conjugate()

```
ex GiNaC::power::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), basis, GiNaC::ex::conjugate(), GiNaC::dynallocate(), GiNaC::basic::ex, exponent, GiNaC::ex::info(), GiNaC::info_flags::integer, and GiNaC::info_flags::positive.

8.123.3.20 real_part()

```
ex GiNaC::power::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::abs(), basis, GiNaC::binomial(), GiNaC::cos(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::exp(), exponent, GiNaC::ex::imag_part(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::ex::is_equal(), GiNaC::log(), GiNaC::info_flags::nonnegative, GiNaC::numer(), GiNaC::pow(), and GiNaC::ex::real_part().

8.123.3.21 imag_part()

```
ex GiNaC::power::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::abs(), basis, GiNaC::binomial(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::exp(), exponent, GiNaC::ex::imag_part(), GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::ex::is_equal(), GiNaC::log(), GiNaC::info_flags::nonnegative, GiNaC::numer(), GiNaC::pow(), GiNaC::ex::real_part(), and GiNaC::sin().

8.123.3.22 archive()

```
void GiNaC::power::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_ex(), basis, and exponent.

8.123.3.23 read_archive()

```
void GiNaC::power::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References basis, exponent, and GiNaC::archive_node::find_ex().

8.123.3.24 derivative()

```
ex GiNaC::power::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for a power.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::_ex_1, basis, GiNaC::ex::diff(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::expair::expair(), exponent, GiNaC::is_a(), GiNaC::log(), and GiNaC::pow().

8.123.3.25 eval_ncmul()

```
ex GiNaC::power::eval_ncmul (
            const exvector & v) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.123.3.26 return_type()

```
unsigned GiNaC::power::return_type () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References basis, and GiNaC::ex::return_type().

8.123.3.27 return_type_tinfo()

```
return_type_t GiNaC::power::return_type_tinfo () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References basis, and GiNaC::ex::return_type_tinfo().

8.123.3.28 expand()

```
ex GiNaC::power::expand (
            unsigned options = 0) const  [override], [protected], [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::_ex_1, GiNaC::are_ex_trivially_equal(), basis, coeff(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::ex::expand(), expand(), expand_add(), expand_mul(), GiNaC::status_flags::expanded, exponent, GiNaC::basic::hold(), GiNaC::info_flags::indefinite, GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::is_a(), GiNaC::is_exactly_a(), GiNaC::is_integer(), mul, GiNaC::info_flags::negative, GiNaC::expairseq::overall_coeff, GiNaC::info_flags::positive, GiNaC::pow(), GiNaC::status_flags::purely_indefinite, GiNaC::add::recombine_pair_to_ex(), GiNaC::mul::recombine_pair_to_ex(), GiNaC::expairseq::seq, GiNaC::basic::setflag(), GiNaC::numeric::to_int(), and GiNaC::numeric::to_long().

Referenced by expand(), expand_add(), and expand_add_2().

8.123.3.29 print_power()

```
void GiNaC::power::print_power (
            const print_context & c,
            const char * powersymbol,
            const char * openbrace,
            const char * closebrace,
            unsigned level) const  [protected]
```

References basis, exponent, precedence(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by do_print_dflt(), do_print_latex(), and do_print_python().

8.123.3.30 do_print_dflt()

```
void GiNaC::power::do_print_dflt (
            const print_dflt & c,
            unsigned level) const  [protected]
```

References GiNaC::_ex1_2, basis, exponent, GiNaC::ex::is_equal(), GiNaC::ex::print(), print_power(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.123.3.31 do_print_latex()

```
void GiNaC::power::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

References GiNaC::_ex1_2, basis, GiNaC::ex_to(), exponent, GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::is_negative(), power(), GiNaC::ex::print(), print_power(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.123.3.32 do_print_csrc()

```
void GiNaC::power::do_print_csrc (
            const print_csrc & c,
            unsigned level) const  [protected]
```

References GiNaC::_ex_1, basis, GiNaC::ex_to(), GiNaC::exp(), exponent, GiNaC::ex::info(), GiNaC::info_flags::integer, GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::ex::print(), GiNaC::print_sym_pow(), GiNaC::print_context::s, and GiNaC::numeric::to_int().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.123.3.33 do_print_python()

```
void GiNaC::power::do_print_python (
            const print_python & c,
            unsigned level) const  [protected]
```

References print_power().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.123.3.34 do_print_python_repr()

```
void GiNaC::power::do_print_python_repr (
            const print_python_repr & c,
            unsigned level) const  [protected]
```

References basis, exponent, GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.123.3.35 do_print_csrc_cl_N()

```
void GiNaC::power::do_print_csrc_cl_N (
            const print_csrc_cl_N & c,
            unsigned level) const  [protected]
```

References GiNaC::_ex_1, basis, exponent, GiNaC::ex::is_equal(), GiNaC::ex::print(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.123.3.36 expand_add()

```
ex GiNaC::power::expand_add (
            const add & a,
            long n,
            unsigned options)  [static], [protected]
```

expand a^n where a is an add and n is a positive integer.

See also

power::expand

References GiNaC::_ex1, GiNaC::_num1_p, basis, GiNaC::binomial(), coeff(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), expand(), expand_add_2(), GiNaC::status_flags::expanded, exponent, GiNaC::factor(), GiNaC::composition_generator::get(), GiNaC::partition_with_zero_parts_generator::get(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::is_pos_integer(), GiNaC::ex::is_zero(), GiNaC::mul::mul(), GiNaC::multinomial_coefficient(), GiNaC::composition_generator::next(), GiNaC::partition_with_zero_parts_generator::next(), GiNaC::expairseq::nops(), GiNaC::numeric::numeric(), GiNaC::expairseq::overall_coeff, GiNaC::pow(), GiNaC::numeric::power(), GiNaC::expairseq::seq, and GiNaC::numeric::to_long().

Referenced by expand().

8.123.3.37 expand_add_2()

```
ex GiNaC::power::expand_add_2 (
            const add & a,
            unsigned options)  [static], [protected]
```

Special case of power::expand_add.

Expands a^2 where a is an add.

See also

power::expand_add

References GiNaC::_ex1, GiNaC::_ex2, GiNaC::_num2_p, basis, GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), expand(), expand_mul(), GiNaC::status_flags::expanded, exponent, GINAC_ASSERT, GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::is_pos_integer(), GiNaC::ex::is_zero(), GiNaC::mul::mul(), GiNaC::numeric::mul(), GiNaC::numeric::mul_dyn(), GiNaC::expairseq::nops(), GiNaC::expairseq::overall_coeff, and GiNaC::expairseq::seq.

Referenced by expand_add().

8.123.3.38 expand_mul()

```
ex GiNaC::power::expand_mul (
              const mul & m,
              const numeric & n,
              unsigned options,
              bool from_expand = false)  [static], [protected]
```

Expand factors of m in m^n where m is a mul and n is an integer.

See also

power::expand

References GiNaC::_ex1, GiNaC::expair::coeff, GiNaC::mul::combine_pair_with_coeff_to_pair(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex::ex(), GiNaC::ex_to(), GiNaC::expand_options::expand_rename_idx, GiNaC::status_flags::expanded, GiNaC::get_all_dummy_indices(), GINAC_ASSERT, GiNaC::info_flags::has_indices, GiNaC::mul::info(), GiNaC::is_exactly_a(), GiNaC::numeric::is_integer(), GiNaC::numeric::is_positive(), GiNaC::numeric::is_zero(), mul, GiNaC::expairseq::overall_coeff, GiNaC::rename_dummy_indices_uniquely(), GiNaC::expairseq::seq, GiNaC::basic::setflag(), and GiNaC::numeric::to_int().

Referenced by eval(), expand(), and expand_add_2().

8.123.4 Friends And Related Symbol Documentation

8.123.4.1 mul

```
friend class mul  [friend]
```

References mul.

Referenced by eval(), expand(), expand_mul(), and mul.

8.123.5 Member Data Documentation

8.123.5.1 basis

```
ex GiNaC::power::basis  [protected]
```

Referenced by archive(), coeff(), conjugate(), degree(), derivative(), do_print_csrc(), do_print_csrc_cl_N(), do_print_dflt(), do_print_latex(), do_print_python_repr(), eval(), evalf(), evalm(), expand(), expand_add(), expand_add_2(), has(), imag_part(), info(), is_polynomial(), ldegree(), map(), normal(), op(), power(), power(), print_power(), read_archive(), real_part(), return_type(), return_type_tinfo(), series(), GiNaC::mul::split_ex_to_pair(), subs(), to_polynomial(), and to_rational().

8.123.5.2 exponent

```
ex GiNaC::power::exponent  [protected]
```

Referenced by archive(), coeff(), conjugate(), degree(), derivative(), do_print_csrc(), do_print_csrc_cl_N(), do_print_dflt(), do_print_latex(), do_print_python_repr(), eval(), evalf(), evalm(), expand(), expand_add(), expand_add_2(), has(), imag_part(), info(), is_polynomial(), ldegree(), map(), normal(), op(), power(), power(), print_power(), read_archive(), real_part(), series(), GiNaC::mul::split_ex_to_pair(), subs(), to_polynomial(), and to_rational().

The documentation for this class was generated from the following files:

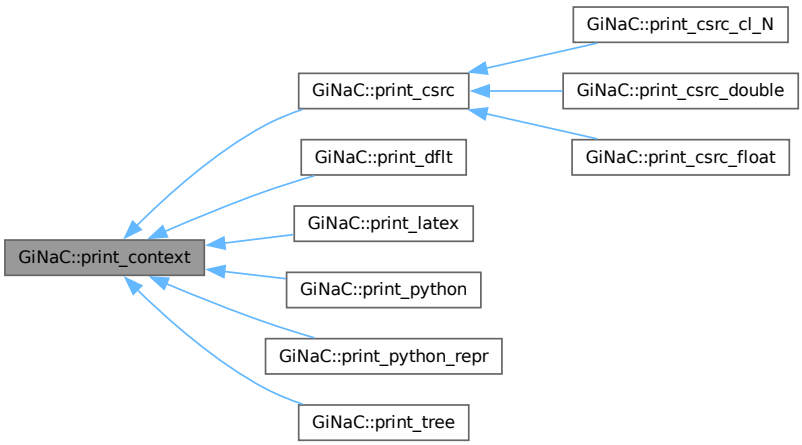
- power.h
- normal.cpp
- power.cpp
- pseries.cpp

8.124 GiNaC::print_context Class Reference

Base class for print_contexts.

```
#include <print.h>
```

Inheritance diagram for GiNaC::print_context:



Public Member Functions

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Public Attributes

- std::ostream & s
  stream to output to
- unsigned options
  option flags

8.124.1 Detailed Description

Base class for print_contexts.

8.124.2 Constructor & Destructor Documentation

8.124.2.1 print_context()

```
GiNaC::print_context::print_context (
            std::ostream & os,
            unsigned options = 0)
```

References options, print_context(), and s.

Referenced by print_context(), GiNaC::print_csrc::print_csrc(), GiNaC::print_dflt::print_dflt(), GiNaC::print_latex::print_latex(), GiNaC::print_python::print_python(), GiNaC::print_python_repr::print_python_repr(), GiNaC::print_tree::print_tree(), and GiNaC::print_tree::print_tree().

8.124.2.2 ~print_context()

```
virtual GiNaC::print_context::~print_context ()  [inline], [virtual]
```

8.124.3 Member Data Documentation

8.124.3.1 s

```
std::ostream& GiNaC::print_context::s
```

stream to output to

Referenced by GiNaC::abs_print_csrc_float(), GiNaC::abs_print_latex(), GiNaC::conjugate_print_latex(), GiNaC::basic::do_print(), GiNaC::basic_log_kernel::do_print(), GiNaC::constant::do_print(), GiNaC::Ebar_kernel::do_print(), GiNaC::Eisenstein_h_kernel::do_print(), GiNaC::Eisenstein_kernel::do_print(), GiNaC::ELi_kernel::do_print(), GiNaC::expairseq::do_print(), GiNaC::fderivative::do_print(), GiNaC::Gt::do_print(), GiNaC::idx::do_print(), GiNaC::integral::do_print(), GiNaC::integration_kernel::do_print(), GiNaC::Kronecker_dtau_kernel::do_print(), GiNaC::Kronecker_dz_kernel::do_print(), GiNaC::matrix::do_print(), GiNaC::modular_form_kernel::do_print(), GiNaC::mul::do_print(), GiNaC::multiple_polylog_kernel::do_print(), GiNaC::relational::do_print(), GiNaC::spinidx::do_print(), GiNaC::symbol::do_print(), GiNaC::symmetry::do_print(), GiNaC::user_defined_kernel::do_print(), GiNaC::varidx::do_print(), GiNaC::wildcard::do_print(), GiNaC::add::do_print_csrc(), GiNaC::fderivative::do_print_csrc(), GiNaC::idx::do_print_csrc(), GiNaC::mul::do_print_csrc(), GiNaC::ncmul::do_print_csrc(), GiNaC::numeric::do_print_csrc(), GiNaC::power::do_print_csrc(), GiNaC::numeric::do_print_csrc_cl_N(), GiNaC::power::do_print_csrc_cl_N(), GiNaC::clifford::do_print_dflt(), GiNaC::power::do_print_dflt(), GiNaC::clifford::do_print_latex(), GiNaC::constant::do_print_latex(), GiNaC::fderivative::do_print_latex(

GiNaC::Gt::do_print_latex(), GiNaC::idx::do_print_latex(), GiNaC::integral::do_print_latex(), GiNaC::matrix::do_print_latex(), GiNaC::mul::do_print_latex(), GiNaC::power::do_print_latex(), GiNaC::spinidx::do_print_latex(), GiNaC::symbol::do_print_latex(), GiNaC::add::do_print_python_repr(), GiNaC::basic::do_print_python_repr(), GiNaC::constant::do_print_python_repr(), GiNaC::container< C >::do_print_python_repr(), GiNaC::matrix::do_print_python_repr(), GiNaC::mul::do_print_python_repr(), GiNaC::numeric::do_print_python_repr(), GiNaC::power::do_print_python_repr(), GiNaC::pseries::do_print_python_repr(), GiNaC::relational::do_print_python_repr(), GiNaC::symbol::do_print_python_repr(), GiNaC::wildcard::do_print_python_repr(), GiNaC::basic::do_print_tree(), GiNaC::clifford::do_print_tree(), GiNaC::constant::do_print_tree(), GiNaC::container< C >::do_print_tr GiNaC::expairseq::do_print_tree(), GiNaC::fderivative::do_print_tree(), GiNaC::idx::do_print_tree(), GiNaC::indexed::do_print_tree(), GiNaC::numeric::do_print_tree(), GiNaC::pseries::do_print_tree(), GiNaC::spinidx::do_print_tree(), GiNaC::symbol::do_print_tree(), GiNaC::symmetry::do_print_tree(), GiNaC::varidx::do_print_tree(), GiNaC::wildcard::do_print_tree(), GiNaC::EllipticE_print_latex(), GiNaC::EllipticK_print_latex(), GiNaC::factorial_print_dflt_latex(), GiNaC::H_print_latex(), GiNaC::imag_part_print_latex(), GiNaC::Li_print_latex(), GiNaC::function::print(), GiNaC::add::print_add(), print_context(), GiNaC::matrix::print_elements(), GiNaC::idx::print_index(), GiNaC::indexed::print_indexed(), GiNaC::print_integer_csrc(), GiNaC::numeric::print_numeric(), GiNaC::print_operator(), GiNaC::mul::print_overall_coeff(), GiNaC::power::print_power(), GiNaC::print_real_cl_N(), GiNaC::print_real_csrc(), GiNaC::print_real_number(), GiNaC::pseries::print_series(), GiNaC::print_sym_pow(), GiNaC::indexed::printindices(), GiNaC::expairseq::printpair(), GiNaC::container< C >::printseq(), GiNaC::expairseq::printseq(), GiNaC::real_part_print_latex(), GiNaC::S_print_latex(), GiNaC::zeta1_print_latex(), and GiNaC::zeta2_print_latex().

8.124.3.2 options

```
unsigned GiNaC::print_context::options
```

option flags

Referenced by GiNaC::get_print_options(), print_context(), GiNaC::idx::print_index(), GiNaC::set_print_context(), and GiNaC::set_print_options().

The documentation for this class was generated from the following files:

- print.h
- print.cpp

8.125 GiNaC::print_context_options Class Reference

This class stores information about a registered print_context class.

```
#include <print.h>
```

Public Member Functions

- print_context_options (const char *n, const char *p, unsigned i)
- const char * get_name () const
- const char * get_parent_name () const
- unsigned get_id () const

Private Attributes

- const char * name

  Class name.
- const char * parent_name

  Name of superclass.
- unsigned id

  ID number (assigned automatically).

8.125.1 Detailed Description

This class stores information about a registered print_context class.

8.125.2 Constructor & Destructor Documentation

8.125.2.1 print_context_options()

```
GiNaC::print_context_options::print_context_options (
            const char * n,
            const char * p,
            unsigned i)  [inline]
```

References id, name, and parent_name.

8.125.3 Member Function Documentation

8.125.3.1 get_name()

```
const char * GiNaC::print_context_options::get_name () const  [inline]
```

References name.

8.125.3.2 get_parent_name()

```
const char * GiNaC::print_context_options::get_parent_name () const  [inline]
```

References parent_name.

8.125.3.3 get_id()

```
unsigned GiNaC::print_context_options::get_id () const  [inline]
```

References id.

Referenced by GiNaC::function::print(), and GiNaC::basic::print_dispatch().

8.125.4 Member Data Documentation

8.125.4.1 name

```
const char* GiNaC::print_context_options::name  [private]
```

Class name.

Referenced by get_name(), and print_context_options().

8.125.4.2 parent_name

```
const char* GiNaC::print_context_options::parent_name  [private]
```

Name of superclass.

Referenced by get_parent_name(), and print_context_options().

8.125.4.3 id

```
unsigned GiNaC::print_context_options::id  [private]
```

ID number (assigned automatically).

Referenced by get_id(), and print_context_options().

The documentation for this class was generated from the following file:

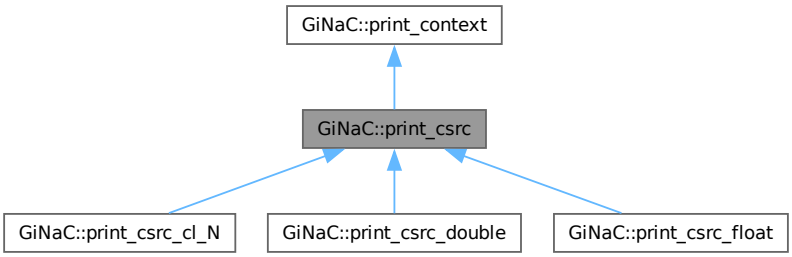
- print.h

8.126 GiNaC::print_csrc Class Reference

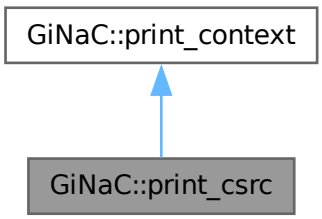
Base context for C source output.

`#include <print.h>`

Inheritance diagram for GiNaC::print_csrc:



Collaboration diagram for GiNaC::print_csrc:



Public Member Functions

- print_csrc (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_context

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Additional Inherited Members

Public Attributes inherited from GiNaC::print_context

- std::ostream & s

  stream to output to
- unsigned options

  option flags

8.126.1 Detailed Description

Base context for C source output.

8.126.2 Constructor & Destructor Documentation

8.126.2.1 print_csrc()

```
GiNaC::print_csrc::print_csrc (
            std::ostream & os,
            unsigned options = 0)
```

References GiNaC::print_context::print_context().

Referenced by GiNaC::print_csrc_cl_N::print_csrc_cl_N(), GiNaC::print_csrc_double::print_csrc_double(), and GiNaC::print_csrc_float::print_csrc_float().

The documentation for this class was generated from the following files:

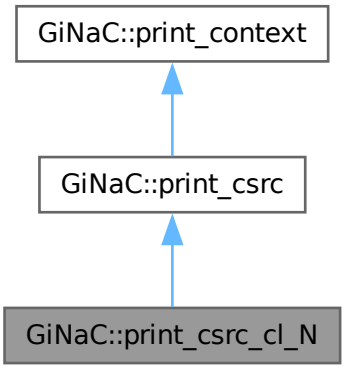
- print.h
- print.cpp

8.127 GiNaC::print_csrc_cl_N Class Reference

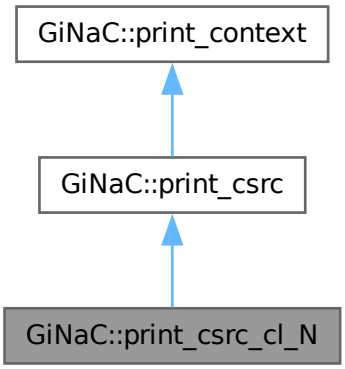
Context for C source output using CLN numbers.

`#include <print.h>`

Inheritance diagram for GiNaC::print_csrc_cl_N:



Collaboration diagram for GiNaC::print_csrc_cl_N:



Public Member Functions

- print_csrc_cl_N (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_csrc

- print_csrc (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_context

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Additional Inherited Members

Public Attributes inherited from GiNaC::print_context

- std::ostream & s

  stream to output to
- unsigned options

  option flags

8.127.1 Detailed Description

Context for C source output using CLN numbers.

8.127.2 Constructor & Destructor Documentation

8.127.2.1 print_csrc_cl_N()

```
GiNaC::print_csrc_cl_N::print_csrc_cl_N (
            std::ostream & os,
            unsigned options = 0)
```

References GiNaC::print_csrc::print_csrc().

Referenced by GiNaC::csrc_cl_N().

The documentation for this class was generated from the following files:

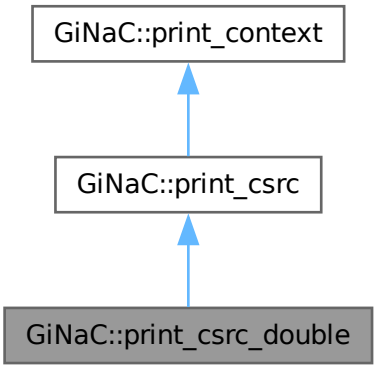
- print.h
- print.cpp

8.128 GiNaC::print_csrc_double Class Reference

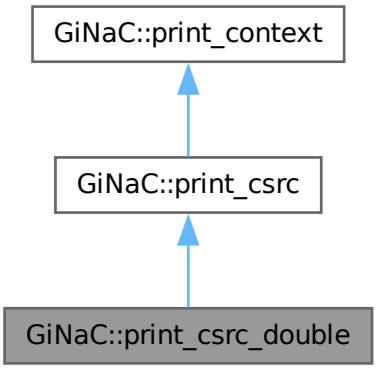
Context for C source output using double precision.

`#include <print.h>`

Inheritance diagram for GiNaC::print_csrc_double:



Collaboration diagram for GiNaC::print_csrc_double:



Public Member Functions

- print_csrc_double (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_csrc

- print_csrc (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_context

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Additional Inherited Members

Public Attributes inherited from GiNaC::print_context

- std::ostream & s

  stream to output to
- unsigned options

  option flags

8.128.1 Detailed Description

Context for C source output using double precision.

8.128.2 Constructor & Destructor Documentation

8.128.2.1 print_csrc_double()

```
GiNaC::print_csrc_double::print_csrc_double (
            std::ostream & os,
            unsigned options = 0)
```

References GiNaC::print_csrc::print_csrc().

Referenced by GiNaC::csrc(), and GiNaC::csrc_double().

The documentation for this class was generated from the following files:

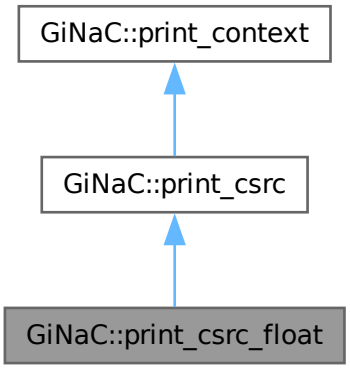
- print.h
- print.cpp

8.129 GiNaC::print_csrc_float Class Reference

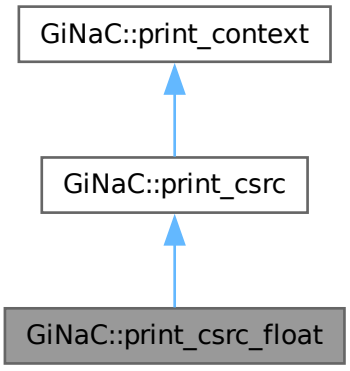
Context for C source output using float precision.

`#include <print.h>`

Inheritance diagram for GiNaC::print_csrc_float:



Collaboration diagram for GiNaC::print_csrc_float:



Public Member Functions

- print_csrc_float (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_csrc

- print_csrc (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_context

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Additional Inherited Members

Public Attributes inherited from GiNaC::print_context

- std::ostream & s

  stream to output to
- unsigned options

  option flags

8.129.1 Detailed Description

Context for C source output using float precision.

8.129.2 Constructor & Destructor Documentation

8.129.2.1 print_csrc_float()

```
GiNaC::print_csrc_float::print_csrc_float (
            std::ostream & os,
            unsigned options = 0)
```

References GiNaC::print_csrc::print_csrc().

Referenced by GiNaC::csrc_float().

The documentation for this class was generated from the following files:

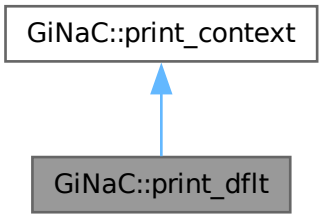
- print.h
- print.cpp

8.130 GiNaC::print_dflt Class Reference

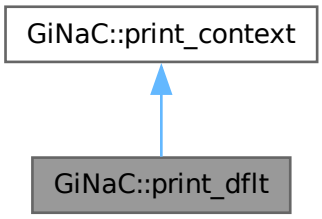
Context for default (ginsh-parsable) output.

`#include <print.h>`

Inheritance diagram for GiNaC::print_dflt:



Collaboration diagram for GiNaC::print_dflt:



Public Member Functions

- print_dflt (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_context

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Additional Inherited Members

Public Attributes inherited from GiNaC::print_context

- std::ostream & s
  stream to output to
- unsigned options
  option flags

8.130.1 Detailed Description

Context for default (ginsh-parsable) output.

8.130.2 Constructor & Destructor Documentation

8.130.2.1 print_dflt()

```
GiNaC::print_dflt::print_dflt (
            std::ostream & os,
            unsigned options = 0)
```

References GiNaC::print_context::print_context().

Referenced by GiNaC::dflt(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(), GiNaC::operator<<(), and GiNaC::set_print_options().

The documentation for this class was generated from the following files:

- print.h
- print.cpp

8.131 GiNaC::print_functor Class Reference

This class represents a print method for a certain algebraic class and print_context type.

```
#include <print.h>
```

Public Member Functions

- print_functor ()
- print_functor (const print_functor &other)
- print_functor (std::unique_ptr< print_functor_impl > impl_)
- template<class T, class C>
  print_functor (void f(const T &, const C &, unsigned))
- template<class T, class C>
  print_functor (void(T::*f)(const C &, unsigned) const)
- print_functor & operator= (const print_functor &other)
- void operator() (const basic &obj, const print_context &c, unsigned level) const
- bool is_valid () const

Private Attributes

- std::unique_ptr< print_functor_impl > impl

8.131.1 Detailed Description

This class represents a print method for a certain algebraic class and print_context type.

Its main purpose is to hide the difference between member functions and nonmember functions behind one unified operator() interface. print_functor has value semantics and acts as a smart pointer (with deep copy) to a class derived from print_functor_impl which implements the actual function call.

8.131.2 Constructor & Destructor Documentation

8.131.2.1 print_functor() [1/5]

```
GiNaC::print_functor::print_functor ()  [inline]
```

References impl.

Referenced by operator=(), and print_functor().

8.131.2.2 print_functor() [2/5]

```
GiNaC::print_functor::print_functor (
            const print_functor & other)  [inline]
```

References impl, and print_functor().

8.131.2.3 print_functor() [3/5]

```
GiNaC::print_functor::print_functor (
            std::unique_ptr< print_functor_impl > impl_)  [inline]
```

References impl.

8.131.2.4 print_functor() [4/5]

```
template<class T, class C>
GiNaC::print_functor::print_functor (
            void  fconst T &, const C &, unsigned)  [inline]
```

References impl.

8.131.2.5 print_functor() [5/5]

```
template<class T, class C>
GiNaC::print_functor::print_functor (
            void(T::* f )(const C &, unsigned) const)  [inline]
```

References impl.

8.131.3 Member Function Documentation

8.131.3.1 operator=()

```
print_functor & GiNaC::print_functor::operator= (
            const print_functor & other)  [inline]
```

References impl, and print_functor().

8.131.3.2 operator()()

```
void GiNaC::print_functor::operator() (
            const basic & obj,
            const print_context & c,
            unsigned level) const  [inline]
```

8.131.3.3 is_valid()

```
bool GiNaC::print_functor::is_valid () const  [inline]
```

References impl.

8.131.4 Member Data Documentation

8.131.4.1 impl

```
std::unique_ptr<print_functor_impl> GiNaC::print_functor::impl  [private]
```

Referenced by is_valid(), operator=(), print_functor(), print_functor(), print_functor(), print_functor(), and print_functor().

The documentation for this class was generated from the following file:

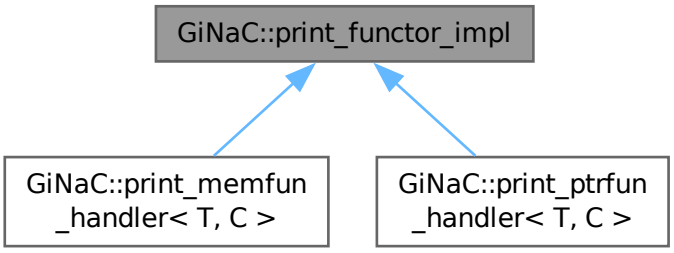
- print.h

8.132 GiNaC::print_functor_impl Class Reference

Base class for print_functor handlers.

`#include <print.h>`

Inheritance diagram for GiNaC::print_functor_impl:



Public Member Functions

- virtual ~print_functor_impl ()
- virtual print_functor_impl * duplicate () const =0
- virtual void operator() (const basic &obj, const print_context &c, unsigned level) const =0

8.132.1 Detailed Description

Base class for print_functor handlers.

8.132.2 Constructor & Destructor Documentation

8.132.2.1 ~print_functor_impl()

`virtual GiNaC::print_functor_impl::~print_functor_impl () [inline], [virtual]`

8.132.3 Member Function Documentation

8.132.3.1 duplicate()

`virtual print_functor_impl * GiNaC::print_functor_impl::duplicate () const [pure virtual]`

Implemented in GiNaC::print_memfun_handler< T, C >, and GiNaC::print_ptrfun_handler< T, C >.

8.132.3.2 operator()()

```
virtual void GiNaC::print_functor_impl::operator() (
             const basic & obj,
             const print_context & c,
             unsigned level) const  [pure virtual]
```

Implemented in GiNaC::print_memfun_handler< T, C >, and GiNaC::print_ptrfun_handler< T, C >.

The documentation for this class was generated from the following file:

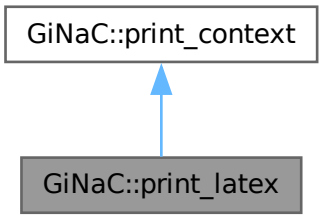
- print.h

8.133 GiNaC::print_latex Class Reference

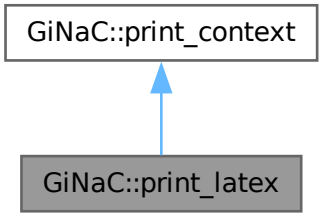
Context for latex-parsable output.

`#include <print.h>`

Inheritance diagram for GiNaC::print_latex:



Collaboration diagram for GiNaC::print_latex:



Public Member Functions

- print_latex (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_context

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Additional Inherited Members

Public Attributes inherited from GiNaC::print_context

- std::ostream & s

  stream to output to
- unsigned options

  option flags

8.133.1 Detailed Description

Context for latex-parsable output.

8.133.2 Constructor & Destructor Documentation

8.133.2.1 print_latex()

```
GiNaC::print_latex::print_latex (
            std::ostream & os,
            unsigned options = 0)
```

References GiNaC::print_context::print_context().

Referenced by GiNaC::latex().

The documentation for this class was generated from the following files:

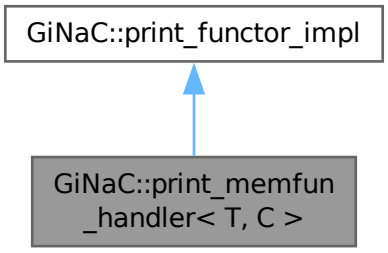
- print.h
- print.cpp

8.134 GiNaC::print_memfun_handler< T, C > Class Template Reference

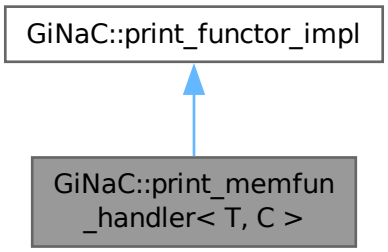
print_functor handler for member functions of class T, context type C

`#include <print.h>`

Inheritance diagram for GiNaC::print_memfun_handler< T, C >:



Collaboration diagram for GiNaC::print_memfun_handler< T, C >:



Public Types

- typedef void(T::*) F(const C &c, unsigned level) const

Public Member Functions

- print_memfun_handler (F f_)
- print_memfun_handler * duplicate () const override
- void operator() (const basic &obj, const print_context &c, unsigned level) const override

Public Member Functions inherited from GiNaC::print_functor_impl

- virtual ~print_functor_impl ()

Private Attributes

- F f

8.134.1 Detailed Description

template<class T, class C>
class GiNaC::print_memfun_handler< T, C >

print_functor handler for member functions of class T, context type C

8.134.2 Member Typedef Documentation

8.134.2.1 F

```
template<class T, class C>
typedef void(T::*) GiNaC::print_memfun_handler< T, C >::F(const C &c, unsigned level) const
```

8.134.3 Constructor & Destructor Documentation

8.134.3.1 print_memfun_handler()

```
template<class T, class C>
GiNaC::print_memfun_handler< T, C >::print_memfun_handler (
            F f_)  [inline]
```

References f.

Referenced by duplicate().

8.134.4 Member Function Documentation

8.134.4.1 duplicate()

```
template<class T, class C>
print_memfun_handler * GiNaC::print_memfun_handler< T, C >::duplicate () const  [inline],
[override], [virtual]
```

Implements GiNaC::print_functor_impl.

References print_memfun_handler().

8.134.4.2 operator()()

```
template<class T, class C>
void GiNaC::print_memfun_handler< T, C >::operator() (
             const basic & obj,
             const print_context & c,
             unsigned level) const  [inline], [override], [virtual]
```

Implements GiNaC::print_functor_impl.

References f.

8.134.5 Member Data Documentation

8.134.5.1 f

```
template<class T, class C>
F GiNaC::print_memfun_handler< T, C >::f  [private]
```

Referenced by operator()(), and print_memfun_handler().

The documentation for this class was generated from the following file:

- print.h

8.135 GiNaC::print_options Class Reference

Flags to control the behavior of a print_context.

```
#include <print.h>
```

Public Types

- enum { print_index_dimensions = 0x0001 }

8.135.1 Detailed Description

Flags to control the behavior of a print_context.

8.135.2 Member Enumeration Documentation

8.135.2.1 anonymous enum

anonymous enum

Enumerator

print_index_dimensions	print the dimensions of indices

The documentation for this class was generated from the following file:

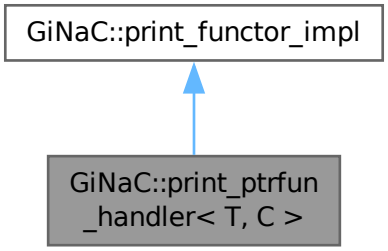
- print.h

8.136 GiNaC::print_ptrfun_handler< T, C > Class Template Reference

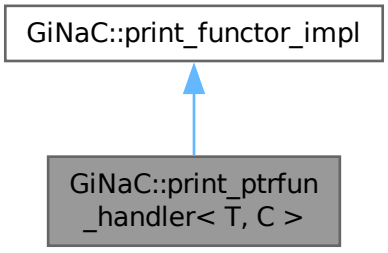
print_functor handler for pointer-to-functions of class T, context type C

#include <print.h>

Inheritance diagram for GiNaC::print_ptrfun_handler< T, C >:



Collaboration diagram for GiNaC::print_ptrfun_handler< T, C >:



Public Types

- typedef void(∗) F(const T &, const C &, unsigned)

Public Member Functions

- print_ptrfun_handler (F f_)
- print_ptrfun_handler ∗ duplicate () const override
- void operator() (const basic &obj, const print_context &c, unsigned level) const override

Public Member Functions inherited from GiNaC::print_functor_impl

- virtual ∼print_functor_impl ()

Private Attributes

- F f

8.136.1 Detailed Description

template<class T, class C>
class GiNaC::print_ptrfun_handler< T, C >

print_functor handler for pointer-to-functions of class T, context type C

8.136.2 Member Typedef Documentation

8.136.2.1 F

```
template<class T, class C>
typedef void(*) GiNaC::print_ptrfun_handler< T, C >::F(const T &, const C &, unsigned)
```

8.136.3 Constructor & Destructor Documentation

8.136.3.1 print_ptrfun_handler()

```
template<class T, class C>
GiNaC::print_ptrfun_handler< T, C >::print_ptrfun_handler (
            F f_)  [inline]
```

References f.

Referenced by duplicate().

8.136.4 Member Function Documentation

8.136.4.1 duplicate()

```
template<class T, class C>
print_ptrfun_handler * GiNaC::print_ptrfun_handler< T, C >::duplicate () const  [inline],
[override], [virtual]
```

Implements GiNaC::print_functor_impl.

References print_ptrfun_handler().

8.136.4.2 operator()()

```
template<class T, class C>
void GiNaC::print_ptrfun_handler< T, C >::operator() (
            const basic & obj,
            const print_context & c,
            unsigned level) const  [inline], [override], [virtual]
```

Implements GiNaC::print_functor_impl.

References f.

8.136.5 Member Data Documentation

8.136.5.1 f

```
template<class T, class C>
F GiNaC::print_ptrfun_handler< T, C >::f  [private]
```

Referenced by operator()(), and print_ptrfun_handler().

The documentation for this class was generated from the following file:

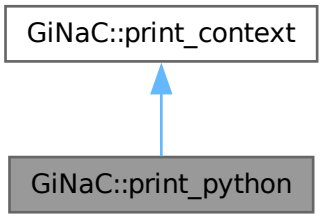
- print.h

8.137 GiNaC::print_python Class Reference

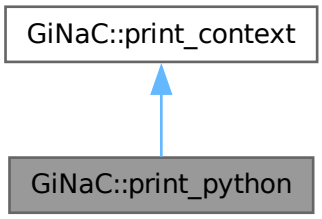
Context for python pretty-print output.

`#include <print.h>`

Inheritance diagram for GiNaC::print_python:



Collaboration diagram for GiNaC::print_python:



Public Member Functions

- print_python (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_context

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Additional Inherited Members

Public Attributes inherited from GiNaC::print_context

- std::ostream & s
  stream to output to
- unsigned options
  option flags

8.137.1 Detailed Description

Context for python pretty-print output.

8.137.2 Constructor & Destructor Documentation

8.137.2.1 print_python()

```
GiNaC::print_python::print_python (
             std::ostream & os,
             unsigned options = 0)
```

References GiNaC::print_context::print_context().

Referenced by GiNaC::python().

The documentation for this class was generated from the following files:

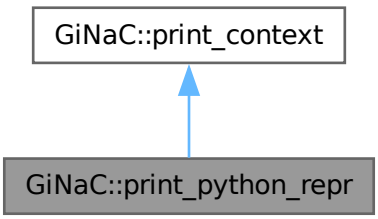
- print.h
- print.cpp

8.138 GiNaC::print_python_repr Class Reference

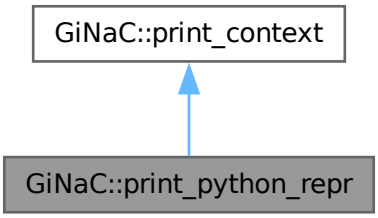
Context for python-parsable output.

```
#include <print.h>
```

Inheritance diagram for GiNaC::print_python_repr:



Collaboration diagram for GiNaC::print_python_repr:



Public Member Functions

- print_python_repr (std::ostream &, unsigned options=0)

Public Member Functions inherited from GiNaC::print_context

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Additional Inherited Members

Public Attributes inherited from GiNaC::print_context

- std::ostream & s
  stream to output to
- unsigned options
  option flags

8.138.1 Detailed Description

Context for python-parsable output.

8.138.2 Constructor & Destructor Documentation

8.138.2.1 print_python_repr()

```
GiNaC::print_python_repr::print_python_repr (
            std::ostream & os,
            unsigned options = 0)
```

References GiNaC::print_context::print_context().

Referenced by GiNaC::python_repr().

The documentation for this class was generated from the following files:

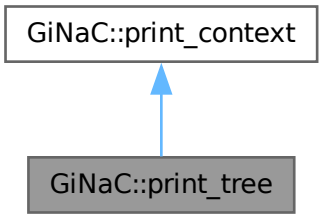
- print.h
- print.cpp

8.139 GiNaC::print_tree Class Reference

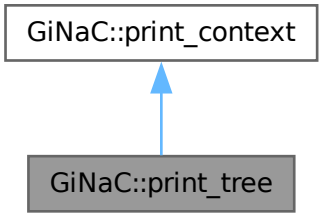
Context for tree-like output for debugging.

`#include <print.h>`

Inheritance diagram for GiNaC::print_tree:



Collaboration diagram for GiNaC::print_tree:



Public Member Functions

- print_tree (unsigned d)
- print_tree (std::ostream &, unsigned options=0, unsigned d=4)

Public Member Functions inherited from GiNaC::print_context

- print_context (std::ostream &, unsigned options=0)
- virtual ~print_context ()

Public Attributes

- const unsigned delta_indent

  size of indentation step

Public Attributes inherited from GiNaC::print_context

- std::ostream & s
  stream to output to
- unsigned options
  option flags

8.139.1 Detailed Description

Context for tree-like output for debugging.

8.139.2 Constructor & Destructor Documentation

8.139.2.1 print_tree() [1/2]

```
GiNaC::print_tree::print_tree (
            unsigned d)
```

References delta_indent, and GiNaC::print_context::print_context().

Referenced by GiNaC::expairseq::is_canonical(), and GiNaC::tree().

8.139.2.2 print_tree() [2/2]

```
GiNaC::print_tree::print_tree (
            std::ostream & os,
            unsigned options = 0,
            unsigned d = 4)
```

References delta_indent, and GiNaC::print_context::print_context().

8.139.3 Member Data Documentation

8.139.3.1 delta_indent

```
const unsigned GiNaC::print_tree::delta_indent
```

size of indentation step

Referenced by GiNaC::basic::do_print_tree(), GiNaC::clifford::do_print_tree(), GiNaC::container< C >::do_print_tree(), GiNaC::expairseq::do_print_tree(), GiNaC::fderivative::do_print_tree(), GiNaC::idx::do_print_tree(), GiNaC::indexed::do_print_tree(), GiNaC::pseries::do_print_tree(), GiNaC::spinidx::do_print_tree(), GiNaC::symmetry::do_print_tree(), GiNaC::varidx::do_print_tree(), GiNaC::expair::print(), print_tree(), and print_tree().

The documentation for this class was generated from the following files:

- print.h
- print.cpp

8.140 GiNaC::archive_node::property Struct Reference

Archived property (data type, name and associated data).

`#include <archive.h>`

Public Member Functions

- property ()
- property (archive_atom n, property_type t, unsigned v)

Public Attributes

- property_type type
  Data type of property.
- archive_atom name
  Name of property.
- unsigned value
  Stored value.

8.140.1 Detailed Description

Archived property (data type, name and associated data).

8.140.2 Constructor & Destructor Documentation

8.140.2.1 property() [1/2]

```
GiNaC::archive_node::property::property ()  [inline]
```

Referenced by GiNaC::archive_node::add_bool(), GiNaC::archive_node::add_ex(), GiNaC::archive_node::add_string(), and GiNaC::archive_node::add_unsigned().

8.140.2.2 property() [2/2]

```
GiNaC::archive_node::property::property (
            archive_atom n,
            property_type t,
            unsigned v)  [inline]
```

References name, type, and value.

8.140.3 Member Data Documentation

8.140.3.1 type

```
property_type GiNaC::archive_node::property::type
```

Data type of property.

Referenced by property().

8.140.3.2 name

```
archive_atom GiNaC::archive_node::property::name
```

Name of property.

Referenced by property().

8.140.3.3 value

```
unsigned GiNaC::archive_node::property::value
```

Stored value.

Referenced by property().

The documentation for this struct was generated from the following file:

- archive.h

8.141 GiNaC::archive_node::property_info Struct Reference

Information about a stored property.

```
#include <archive.h>
```

Public Member Functions

- property_info ()
- property_info (property_type t, const std::string &n, unsigned c=1)

Public Attributes

- property_type type
  Data type of property.
- std::string name
  Name of property.
- unsigned count
  Number of occurrences.

8.141.1 Detailed Description

Information about a stored property.

A vector of these structures is returned by get_properties().

See also

get_properties

8.141.2 Constructor & Destructor Documentation

8.141.2.1 property_info() [1/2]

```
GiNaC::archive_node::property_info::property_info ()  [inline]
```

Referenced by GiNaC::archive_node::get_properties().

8.141.2.2 property_info() [2/2]

```
GiNaC::archive_node::property_info::property_info (
            property_type t,
            const std::string & n,
            unsigned c = 1)  [inline]
```

References count, name, and type.

8.141.3 Member Data Documentation

8.141.3.1 type

```
property_type GiNaC::archive_node::property_info::type
```

Data type of property.

Referenced by property_info().

8.141.3.2 name

```
std::string GiNaC::archive_node::property_info::name
```

Name of property.

Referenced by property_info().

8.141.3.3 count

```
unsigned GiNaC::archive_node::property_info::count
```

Number of occurrences.

Referenced by property_info().

The documentation for this struct was generated from the following file:

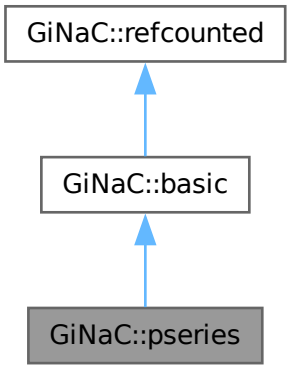
- archive.h

8.142 GiNaC::pseries Class Reference

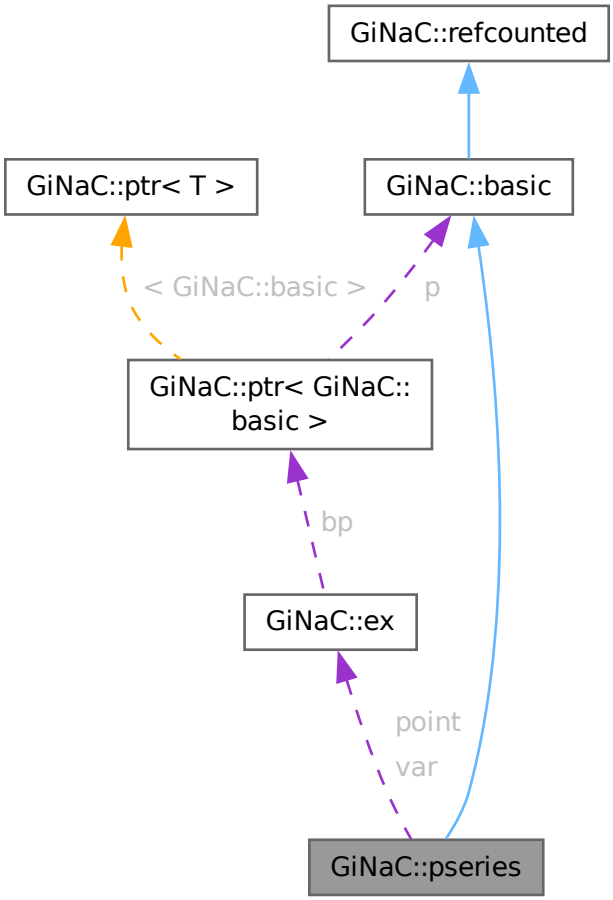
This class holds a extended truncated power series (positive and negative integer powers).

```
#include <pseries.h>
```

Inheritance diagram for GiNaC::pseries:



Collaboration diagram for GiNaC::pseries:



Public Member Functions

- pseries (const ex &rel_, const epvector &ops_)

  Construct pseries from a vector of coefficients and powers.
- pseries (const ex &rel_, epvector &&ops_)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- size_t nops () const override

  Return the number of operands including a possible order term.
- ex op (size_t i) const override

  Return the ith term in the series when represented as a sum.
- int degree (const ex &s) const override

  Return degree of highest power of the series.
- int ldegree (const ex &s) const override

  Return degree of lowest power of the series.
- ex coeff (const ex &s, int n=1) const override

Return coefficient of degree n in power series if s is the expansion variable.

- ex collect (const ex &s, bool distributed=false) const override

  Does nothing.
- ex eval () const override

  Perform coefficient-wise automatic term rewriting rules in this class.
- ex evalf () const override

  Evaluate coefficients numerically.
- ex series (const relational &r, int order, unsigned options=0) const override

  Re-expansion of a pseries object.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const override

  Implementation of ex::normal() for pseries.
- ex expand (unsigned options=0) const override

  Implementation of ex::expand() for a power series.
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- ex eval_integ () const override

  Evaluate integrals, if result is known.
- ex evalm () const override

  Evaluate sums, products and integer powers of matrices.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- ex get_var () const

  Get the expansion variable.
- ex get_point () const

  Get the expansion point.
- ex convert_to_poly (bool no_order=false) const

  Convert the pseries object to an ordinary polynomial.
- bool is_compatible_to (const pseries &other) const

  Check whether series is compatible to another series (expansion variable and point are the same.
- bool is_zero () const

  Check whether series has the value zero.
- bool is_terminating () const

  Returns true if there is no order term, i.e.
- ex coeffop (size_t i) const

  Get coefficients and exponents.
- ex exponop (size_t i) const
- ex add_series (const pseries &other) const

  Add one series object to another, producing a pseries object that represents the sum.
- ex mul_const (const numeric &other) const

  Multiply a pseries object with a numeric constant, producing a pseries object that represents the product.
- ex mul_series (const pseries &other) const

  Multiply one pseries object to another, producing a pseries object that represents the product.
- ex power_const (const numeric &p, int deg) const

  Compute the p-th power of a series.
- pseries shift_exponents (int deg) const

  Return a new pseries object with the powers shifted by deg.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*

- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.

- virtual basic * duplicate () const

  Create a clone of this object on the heap.

- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.

- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.

- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.

- virtual bool info (unsigned inf) const

  Information about the object.

- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.

- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.

- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.

- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().

- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.

- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().

- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.

- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.

- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.

- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.

- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const

- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for a power series.
- void print_series (const print_context &c, const char *openbrace, const char *closebrace, const char *mul_sym, const char *pow_sym, unsigned level) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python (const print_python &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

Returns true if two objects of same type are equal.

- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- epvector seq

  Vector of {coefficient, power} pairs.
- ex var

  Series variable (holds a symbol).
- ex point

  Expansion point.

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.142.1 Detailed Description

This class holds a extended truncated power series (positive and negative integer powers).

It consists of expression coefficients (only non-zero coefficients are stored), an expansion variable and an expansion point. Other classes must provide members to convert into this type.

8.142.2 Constructor & Destructor Documentation

8.142.2.1 pseries() [1/2]

```
GiNaC::pseries::pseries (
            const ex & rel_,
            const epvector & ops_)
```

Construct pseries from a vector of coefficients and powers.

expair.rest holds the coefficient, expair.coeff holds the power. The powers must be integers (positive or negative) and in ascending order; the last coefficient can be Order(_ex1) to represent a truncated, non-terminating series.

Parameters

*rel_*	expansion variable and point (must hold a relational)
*ops_*	vector of {coefficient, power} pairs (coefficient must not be zero)

References GiNaC::basic::ex, GINAC_ASSERT, GiNaC::is_a(), GiNaC::ex::lhs(), point, GiNaC::ex::rhs(), seq, and var.

Referenced by add_series(), GiNaC::atan_series(), GiNaC::atanh_series(), GiNaC::csgn_series(), derivative(), GiNaC::EllipticE_series(), GiNaC::EllipticK_series(), GiNaC::eta_series(), GiNaC::H_series(), is_compatible_to(), GiNaC::Li2_series(), GiNaC::Li_series(), GiNaC::log_series(), mul_const(), mul_series(), normal(), GiNaC::Order_series(), power_const(), GiNaC::S_series(), GiNaC::basic::series(), GiNaC::power::series(), series(), GiNaC::symbol::series(), shift_exponents(), and GiNaC::step_series().

8.142.2.2 pseries() [2/2]

```
GiNaC::pseries::pseries (
            const ex & rel_,
            epvector && ops_)
```

References GiNaC::basic::ex, GINAC_ASSERT, GiNaC::is_a(), GiNaC::ex::lhs(), point, GiNaC::ex::rhs(), seq, and var.

8.142.3 Member Function Documentation

8.142.3.1 precedence()

```
unsigned GiNaC::pseries::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

Referenced by print_series().

8.142.3.2 nops()

```
size_t GiNaC::pseries::nops () const  [override], [virtual]
```

Return the number of operands including a possible order term.

Reimplemented from GiNaC::basic.

References seq.

Referenced by coeffop(), exponop(), and GiNaC::log_series().

8.142.3.3 op()

```
ex GiNaC::pseries::op (
            size_t i) const  [override], [virtual]
```

Return the ith term in the series when represented as a sum.

Reimplemented from GiNaC::basic.

References coeff(), GiNaC::basic::ex, GiNaC::is_order_function(), point, GiNaC::pow(), seq, and var.

8.142.3.4 degree()

```
int GiNaC::pseries::degree (
            const ex & s) const  [override], [virtual]
```

Return degree of highest power of the series.

This is usually the exponent of the Order term. If s is not the expansion variable of the series, the series is examined termwise.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::ex::is_equal(), seq, and var.

Referenced by mul_series(), power_const(), and series().

8.142.3.5 ldegree()

```
int GiNaC::pseries::ldegree (
            const ex & s) const  [override], [virtual]
```

Return degree of lowest power of the series.

This is usually the exponent of the leading term. If s is not the expansion variable of the series, the series is examined termwise. If s is the expansion variable but the expansion point is not zero the series is not expanded to find the degree. I.e.: (1-x) + (1-x)^2 + Order((1-x)^3) has ldegree(x) 1, not 0.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::ex::is_equal(), seq, and var.

Referenced by GiNaC::log_series(), mul_series(), and power_const().

8.142.3.6 coeff()

```
ex GiNaC::pseries::coeff (
            const ex & s,
            int n = 1) const  [override], [virtual]
```

Return coefficient of degree n in power series if s is the expansion variable.

If the expansion point is nonzero, by definition the n=1 coefficient in s of a+b∗(s-z)+c∗(s-z)^2+Order((s-z)^3) is b (assuming the expansion took place in the s in the first place). If s is not the expansion variable, an attempt is made to convert the series to a polynomial and return the corresponding coefficient from there.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::ex::coeff(), coeff(), convert_to_poly(), GiNaC::basic::ex, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::ex::is_equal(), GiNaC::is_exactly_a(), GiNaC::numeric::numeric(), seq, and var.

Referenced by coeff(), GiNaC::log_series(), mul_series(), op(), power_const(), and read_archive().

8.142.3.7 collect()

```
ex GiNaC::pseries::collect (
            const ex & s,
            bool distributed = false) const  [override], [virtual]
```

Does nothing.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.142.3.8 eval()

```
ex GiNaC::pseries::eval () const  [override], [virtual]
```

Perform coefficient-wise automatic term rewriting rules in this class.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, and GiNaC::basic::hold().

8.142.3.9 evalf()

```
ex GiNaC::pseries::evalf () const  [override], [virtual]
```

Evaluate coefficients numerically.

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate(), GiNaC::status_flags::evaluated, GiNaC::basic::ex, GiNaC::expair::expair(), point, GiNaC::relational::relational(), seq, and var.

8.142.3.10 series()

```
ex GiNaC::pseries::series (
            const relational & r,
            int order,
            unsigned options = 0) const  [override], [virtual]
```

Re-expansion of a pseries object.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, convert_to_poly(), degree(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, GiNaC::is_a(), GiNaC::ex::is_equal(), GiNaC::relational::lhs(), point, pseries(), GiNaC::relational::rhs(), seq, GiNaC::ex::series(), and var.

8.142.3.11 subs()

```
ex GiNaC::pseries::subs (
             const exmap & m,
             unsigned options = 0) const  [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References convert_to_poly(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::expair::expair(), point, GiNaC::relational::relational(), seq, GiNaC::ex::subs(), and var.

8.142.3.12 normal()

```
ex GiNaC::pseries::normal (
             exmap & repl,
             exmap & rev_lookup,
             lst & modifier) const  [override], [virtual]
```

Implementation of ex::normal() for pseries.

It normalizes each coefficient and replaces the series by a temporary symbol.

See also

ex::normal

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::ex::coeff(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::ex::is_zero(), point, pseries(), GiNaC::relational::relational(), GiNaC::replace_with_symbol(), seq, and var.

8.142.3.13 expand()

```
ex GiNaC::pseries::expand (
             unsigned options = 0) const  [override], [virtual]
```

Implementation of ex::expand() for a power series.

It expands all the terms individually and returns the resulting series as a new pseries.

Reimplemented from GiNaC::basic.

References GiNaC::ex::coeff(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::status_flags::expanded, GiNaC::ex::is_zero(), point, GiNaC::relational::relational(), seq, and var.

8.142.3.14 conjugate()

```
ex GiNaC::pseries::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), GiNaC::ex::conjugate(), GiNaC::conjugateepvector(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex::info(), point, GiNaC::info_flags::real, seq, and var.

8.142.3.15 real_part()

```
ex GiNaC::pseries::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::ex::info(), point, GiNaC::info_flags::real, GiNaC::ex::real_part(), seq, and var.

8.142.3.16 imag_part()

```
ex GiNaC::pseries::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::ex::info(), point, GiNaC::info_flags::real, GiNaC::ex::real_part(), seq, and var.

8.142.3.17 eval_integ()

```
ex GiNaC::pseries::eval_integ () const  [override], [virtual]
```

Evaluate integrals, if result is known.

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), GiNaC::ex::coeff(), GiNaC::dynallocate(), GiNaC::ex::eval_integ(), GiNaC::basic::ex, GiNaC::expair::expair(), point, seq, and var.

8.142.3.18 evalm()

```
ex GiNaC::pseries::evalm () const  [override], [virtual]
```

Evaluate sums, products and integer powers of matrices.

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), GiNaC::ex::coeff(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::ex::is_zero(), point, seq, and var.

8.142.3.19 archive()

```
void GiNaC::pseries::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_ex(), point, seq, and var.

8.142.3.20 read_archive()

```
void GiNaC::pseries::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References coeff(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::archive_node::find_ex(), GiNaC::archive_node::find_ex_by_loc
GiNaC::archive_node::find_property_range(), point, seq, and var.

8.142.3.21 derivative()

```
ex GiNaC::pseries::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for a power series.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::ex::coeff(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::is_order_function(), GiNaC::ex::is_zero(), point, pseries(), GiNaC::relational::relational(), seq, and var.

8.142.3.22 get_var()

```
ex GiNaC::pseries::get_var () const  [inline]
```

Get the expansion variable.

References GiNaC::basic::ex, and var.

8.142.3.23 get_point()

```
ex GiNaC::pseries::get_point () const  [inline]
```

Get the expansion point.

References GiNaC::basic::ex, and point.

8.142.3.24 convert_to_poly()

```
ex GiNaC::pseries::convert_to_poly (
            bool no_order = false) const
```

Convert the pseries object to an ordinary polynomial.

Parameters

no_order	flag: discard higher order terms

References GiNaC::ex::coeff(), GiNaC::basic::ex, GiNaC::is_order_function(), point, GiNaC::pow(), seq, and var.

Referenced by coeff(), series(), and subs().

8.142.3.25 is_compatible_to()

```
bool GiNaC::pseries::is_compatible_to (
            const pseries & other) const  [inline]
```

Check whether series is compatible to another series (expansion variable and point are the same.

References GiNaC::ex::is_equal(), point, pseries(), and var.

Referenced by add_series(), and mul_series().

8.142.3.26 is_zero()

```
bool GiNaC::pseries::is_zero () const  [inline]
```

Check whether series has the value zero.

References seq.

Referenced by power_const().

8.142.3.27 is_terminating()

```
bool GiNaC::pseries::is_terminating () const
```

Returns true if there is no order term, i.e.

the series terminates and false otherwise.

References GiNaC::is_order_function(), and seq.

Referenced by GiNaC::is_terminating(), and GiNaC::log_series().

8.142.3.28 coeffop()

```
ex GiNaC::pseries::coeffop (
             size_t i) const
```

Get coefficients and exponents.

References GiNaC::basic::ex, nops(), and seq.

8.142.3.29 exponop()

```
ex GiNaC::pseries::exponop (
             size_t i) const
```

References GiNaC::basic::ex, nops(), and seq.

8.142.3.30 add_series()

```
ex GiNaC::pseries::add_series (
             const pseries & other) const
```

Add one series object to another, producing a pseries object that represents the sum.

Parameters

other	pseries object to add with

Returns

the sum as a pseries

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), is_compatible_to(), GiNaC::is_order_function(), GiNaC::ex::is_zero(), GiNaC::numeric::numeric(), point, pseries(), GiNaC::relational::relational(), seq, and var.

8.142.3.31 mul_const()

```
ex GiNaC::pseries::mul_const (
             const numeric & other) const
```

Multiply a pseries object with a numeric constant, producing a pseries object that represents the product.

Parameters

other	constant to multiply with

Returns

the product as a pseries

References GiNaC::numeric::coeff(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::is_order_function(), point, pseries(), GiNaC::relational::relational(), seq, and var.

Referenced by GiNaC::mul::series().

8.142.3.32 mul_series()

```
ex GiNaC::pseries::mul_series (
            const pseries & other) const
```

Multiply one pseries object to another, producing a pseries object that represents the product.

Parameters

other	pseries object to multiply with

Returns

the product as a pseries

References GiNaC::_ex0, GiNaC::_ex1, coeff(), degree(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), is_compatible_to(), GiNaC::ex::is_equal(), GiNaC::is_order_function(), GiNaC::ex::is_zero(), ldegree(), GiNaC::numeric::numeric(), point, pseries(), GiNaC::relational::relational(), seq, and var.

Referenced by GiNaC::mul::series().

8.142.3.33 power_const()

```
ex GiNaC::pseries::power_const (
            const numeric & p,
            int deg) const
```

Compute the p-th power of a series.

Parameters

p	power to compute
deg	truncation order of series calculation

References GiNaC::_ex0, GiNaC::_ex1, coeff(), degree(), GiNaC::dynallocate(), GiNaC::basic::ex, GiNaC::expair::expair(), GiNaC::is_integer(), GiNaC::numeric::is_negative(), GiNaC::is_order_function(), GiNaC::numeric::is_pos_integer(), GiNaC::numeric::is_zero(), is_zero(), ldegree(), point, GiNaC::pole_error::pole_error(), GiNaC::pow(), pseries(), GiNaC::numeric::real(), GiNaC::relational::relational(), seq, GiNaC::to_int(), and var.

8.142.3.34 shift_exponents()

```
pseries GiNaC::pseries::shift_exponents (
            int deg) const
```

Return a new pseries object with the powers shifted by deg.

References point, pseries(), GiNaC::relational::relational(), seq, and var.

8.142.3.35 print_series()

```
void GiNaC::pseries::print_series (
             const print_context & c,
             const char * openbrace,
             const char * closebrace,
             const char * mul_sym,
             const char * pow_sym,
             unsigned level) const  [protected]
```

References GiNaC::_ex1, GiNaC::ex::coeff(), GiNaC::is_order_function(), GiNaC::ex::is_zero(), GiNaC::info_flags::negative, GiNaC::info_flags::numeric, point, GiNaC::info_flags::positive, GiNaC::pow(), precedence(), GiNaC::basic::print(), GiNaC::ex::print(), GiNaC::print_context::s, seq, and var.

Referenced by do_print(), do_print_latex(), and do_print_python().

8.142.3.36 do_print()

```
void GiNaC::pseries::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References print_series().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.142.3.37 do_print_latex()

```
void GiNaC::pseries::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

References print_series().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.142.3.38 do_print_tree()

```
void GiNaC::pseries::do_print_tree (
             const print_tree & c,
             unsigned level) const  [protected]
```

References GiNaC::print_tree::delta_indent, GiNaC::basic::flags, GiNaC::basic::hashvalue, point, GiNaC::ex::print(), GiNaC::print_context::s, seq, and var.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.142.3.39 do_print_python()

```
void GiNaC::pseries::do_print_python (
            const print_python & c,
            unsigned level) const  [protected]
```

References print_series().

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.142.3.40 do_print_python_repr()

```
void GiNaC::pseries::do_print_python_repr (
            const print_python_repr & c,
            unsigned level) const  [protected]
```

References point, GiNaC::ex::print(), GiNaC::print_context::s, seq, and var.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.142.4 Member Data Documentation

8.142.4.1 seq

```
epvector GiNaC::pseries::seq  [protected]
```

Vector of {coefficient, power} pairs.

Referenced by add_series(), archive(), coeff(), coeffop(), conjugate(), convert_to_poly(), degree(), derivative(), do_print_python_repr(), do_print_tree(), eval_integ(), evalf(), evalm(), expand(), exponop(), imag_part(), is_terminating(), is_zero(), ldegree(), mul_const(), mul_series(), nops(), normal(), op(), power_const(), print_series(), pseries(), pseries(), read_archive(), real_part(), series(), shift_exponents(), and subs().

8.142.4.2 var

```
ex GiNaC::pseries::var  [protected]
```

Series variable (holds a symbol).

Referenced by add_series(), archive(), coeff(), conjugate(), convert_to_poly(), degree(), derivative(), do_print_python_repr(), do_print_tree(), eval_integ(), evalf(), evalm(), expand(), get_var(), imag_part(), is_compatible_to(), ldegree(), mul_const(), mul_series(), normal(), op(), power_const(), print_series(), pseries(), pseries(), read_archive(), real_part(), series(), shift_exponents(), and subs().

8.142.4.3 point

```
ex GiNaC::pseries::point  [protected]
```

Expansion point.

Referenced by add_series(), archive(), conjugate(), convert_to_poly(), derivative(), do_print_python_repr(), do_print_tree(), eval_integ(), evalf(), evalm(), expand(), get_point(), imag_part(), is_compatible_to(), mul_const(), mul_series(), normal(), op(), power_const(), print_series(), pseries(), pseries(), read_archive(), real_part(), series(), shift_exponents(), and subs().

The documentation for this class was generated from the following files:

- pseries.h
- normal.cpp
- pseries.cpp

8.143 GiNaC::psi1_SERIAL Class Reference

Polylogarithm and multiple polylogarithm.

```
#include <inifcns.h>
```

Static Public Attributes

- static unsigned serial

8.143.1 Detailed Description

Polylogarithm and multiple polylogarithm.

Nielsen's generalized polylogarithm. Harmonic polylogarithm. Gamma-function. Beta-function. Psi-function (aka digamma-function).

8.143.2 Member Data Documentation

8.143.2.1 serial

```
unsigned GiNaC::psi1_SERIAL::serial  [static]
```

Initial value:

```
                        =
function::register_new(function_options("psi", 1).
                       eval_func(psi1_eval).
                       evalf_func(psi1_evalf).
                       derivative_func(psi1_deriv).
                       series_func(psi1_series).
                       latex_name("\\psi").
                       overloaded(2))
```

Referenced by GiNaC::psi().

The documentation for this class was generated from the following files:

- inifcns.h
- inifcns_gamma.cpp

8.144 GiNaC::psi2_SERIAL Class Reference

Derivatives of Psi-function (aka polygamma-functions).

`#include <inifcns.h>`

Static Public Attributes

- static unsigned serial

8.144.1 Detailed Description

Derivatives of Psi-function (aka polygamma-functions).

8.144.2 Member Data Documentation

8.144.2.1 serial

`unsigned GiNaC::psi2_SERIAL::serial [static]`

Initial value:

```
                        =
function::register_new(function_options("psi", 2).
                       eval_func(psi2_eval).
                       evalf_func(psi2_evalf).
                       derivative_func(psi2_deriv).
                       series_func(psi2_series).
                       latex_name("\\psi").
                       overloaded(2))
```

Referenced by GiNaC::psi().

The documentation for this class was generated from the following files:

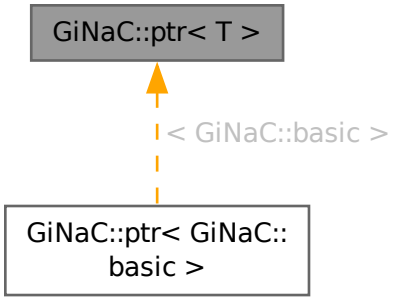
- inifcns.h
- inifcns_gamma.cpp

8.145 GiNaC::ptr< T > Class Template Reference

Class of (intrusively) reference-counted pointers that support copy-on-write semantics.

`#include <ptr.h>`

Inheritance diagram for GiNaC::ptr< T >:



Public Member Functions

- ptr (T *t) noexcept

  Bind ptr to newly created object, start reference counting.
- ptr (T &t) noexcept

  Bind ptr to existing reference-counted object.
- ptr (const ptr &other) noexcept
- ~ptr ()
- ptr & operator= (const ptr &other)
- T & operator* () const noexcept
- T * operator-> () const noexcept
- void makewritable ()

  Announce your intention to modify the object bound to this ptr.
- void swap (ptr &other) noexcept

  Swap the bound object of this ptr with another ptr.
- template<class U>
  bool operator== (const ptr< U > &rhs) const noexcept
- template<class U>
  bool operator!= (const ptr< U > &rhs) const noexcept

Private Attributes

- T * p

Friends

- struct std::less< ptr< T > >
- T * get_pointer (const ptr &x) noexcept
- template<class U>
  bool operator== (const ptr &lhs, const U *rhs) noexcept
- template<class U>
  bool operator!= (const ptr &lhs, const U *rhs) noexcept
- template<class U>
  bool operator== (const U *lhs, const ptr &rhs) noexcept
- template<class U>
  bool operator!= (const U *lhs, const ptr &rhs) noexcept
- std::ostream & operator<< (std::ostream &os, const ptr< T > &rhs)

8.145.1 Detailed Description

template<class T>
class GiNaC::ptr< T >

Class of (intrusively) reference-counted pointers that support copy-on-write semantics.

Requirements for T: must support the refcounted interface (usually by being derived from refcounted) T* T::↩ duplicate() member function (only if makewriteable() is used)

8.145.2 Constructor & Destructor Documentation

8.145.2.1 ptr() [1/3]

```
template<class T>
GiNaC::ptr< T >::ptr (
            T * t)  [inline], [noexcept]
```

Bind ptr to newly created object, start reference counting.

References GINAC_ASSERT, and p.

Referenced by GiNaC::ex::construct_from_basic(), get_pointer, operator!=, operator!=(), operator!=, GiNaC::pointer_to_map_function GiNaC::pointer_to_map_function_1arg< T1 >::operator()(), GiNaC::pointer_to_map_function_2args< T1, T2 >::operator()(), GiNaC::pointer_to_map_function_3args< T1, T2, T3 >::operator()(), operator<<, operator=(), operator==, operator==(), operator==, ptr(), and swap().

8.145.2.2 ptr() [2/3]

```
template<class T>
GiNaC::ptr< T >::ptr (
            T & t)  [inline], [explicit], [noexcept]
```

Bind ptr to existing reference-counted object.

References p.

8.145.2.3 ptr() [3/3]

```
template<class T>
GiNaC::ptr< T >::ptr (
            const ptr< T > & other)  [inline], [noexcept]
```

References p, and ptr().

8.145.2.4 ~ptr()

```
template<class T>
GiNaC::ptr< T >::~ptr ()  [inline]
```

References p.

8.145.3 Member Function Documentation

8.145.3.1 operator=()

```
template<class T>
ptr & GiNaC::ptr< T >::operator= (
            const ptr< T > & other)  [inline]
```

References p, and ptr().

8.145.3.2 operator*()

```
template<class T>
T & GiNaC::ptr< T >::operator* () const  [inline], [noexcept]
```

References p.

8.145.3.3 operator->()

```
template<class T>
T * GiNaC::ptr< T >::operator-> () const  [inline], [noexcept]
```

References p.

8.145.3.4 makewritable()

```
template<class T>
void GiNaC::ptr< T >::makewritable ()  [inline]
```

Announce your intention to modify the object bound to this ptr.

This ensures that the object is not shared by any other ptrs.

References p.

8.145.3.5 swap()

```
template<class T>
void GiNaC::ptr< T >::swap (
            ptr< T > & other)  [inline], [noexcept]
```

Swap the bound object of this ptr with another ptr.

References p, and ptr().

8.145.3.6 operator==()

```
template<class T>
template<class U>
bool GiNaC::ptr< T >::operator== (
            const ptr< U > & rhs) const  [inline], [noexcept]
```

References get_pointer, p, ptr(), and GiNaC::rhs().

8.145.3.7 operator"!=()

```
template<class T>
template<class U>
bool GiNaC::ptr< T >::operator!= (
            const ptr< U > & rhs) const  [inline], [noexcept]
```

References get_pointer, p, ptr(), and GiNaC::rhs().

8.145.4 Friends And Related Symbol Documentation

8.145.4.1 std::less< ptr< T > >

```
template<class T>
friend struct std::less< ptr< T > >  [friend]
```

8.145.4.2 get_pointer

```
template<class T>
T * get_pointer (
            const ptr< T > & x)  [friend]
```

References ptr().

Referenced by operator!=(), and operator==().

8.145.4.3 operator== [1/2]

```
template<class T>
template<class U>
bool operator== (
            const ptr< T > & lhs,
            const U * rhs)  [friend]
```

References GiNaC::lhs(), ptr(), and GiNaC::rhs().

8.145.4.4 operator"!= [1/2]

```
template<class T>
template<class U>
bool operator!= (
            const ptr< T > & lhs,
            const U * rhs)  [friend]
```

References GiNaC::lhs(), ptr(), and GiNaC::rhs().

8.145.4.5 operator== [2/2]

```
template<class T>
template<class U>
bool operator== (
            const U * lhs,
            const ptr< T > & rhs)  [friend]
```

References GiNaC::lhs(), ptr(), and GiNaC::rhs().

8.145.4.6 operator"!= [2/2]

```
template<class T>
template<class U>
bool operator!= (
            const U * lhs,
            const ptr< T > & rhs)  [friend]
```

References GiNaC::lhs(), ptr(), and GiNaC::rhs().

8.145.4.7 operator<<

```
template<class T>
std::ostream & operator<< (
            std::ostream & os,
            const ptr< T > & rhs)  [friend]
```

References ptr(), and GiNaC::rhs().

8.145.5 Member Data Documentation

8.145.5.1 p

```
template<class T>
T* GiNaC::ptr< T >::p  [private]
```

Referenced by makewritable(), operator!=(), operator∗(), operator->(), operator=(), operator==(), ptr(), ptr(), ptr(), swap(), and ∼ptr().

The documentation for this class was generated from the following file:

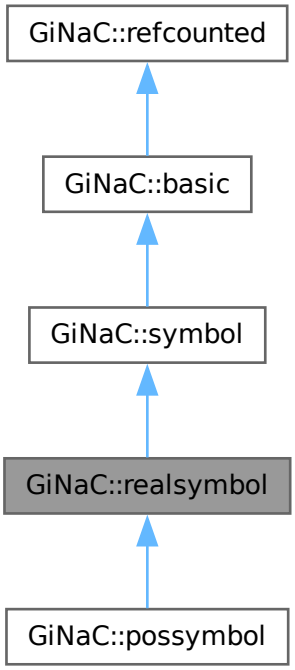
- ptr.h

8.146 GiNaC::realsymbol Class Reference

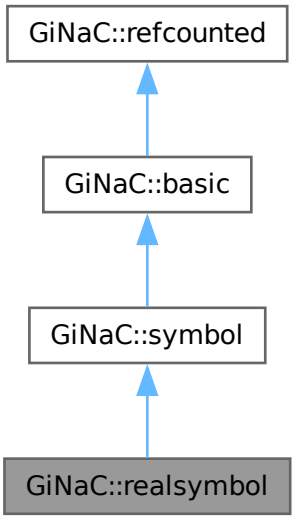
Specialization of symbol to real domain.

```
#include <symbol.h>
```

Inheritance diagram for GiNaC::realsymbol:



Collaboration diagram for GiNaC::realsymbol:



Public Member Functions

- realsymbol ()
- realsymbol (const std::string &initname)
- realsymbol (const std::string &initname, const std::string &texname)
- unsigned get_domain () const override
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- realsymbol * duplicate () const override

  Create a clone of this object on the heap.

Public Member Functions inherited from GiNaC::symbol

- symbol (const std::string &initname)
- symbol (const std::string &initname, const std::string &texname)
- bool info (unsigned inf) const override

  Information about the object.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex evalf () const override

  Evaluate object numerically.
- ex series (const relational &s, int order, unsigned options=0) const override

  Implementation of ex::series() for symbols.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const override

  Implementation of ex::normal() for symbols.
- ex to_rational (exmap &repl) const override

  Implementation of ex::to_rational() for symbols.
- ex to_polynomial (exmap &repl) const override

  Implementation of ex::to_polynomial() for symbols.
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- bool is_polynomial (const ex &var) const override

  Check whether this is a polynomial in the given variables.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- void set_name (const std::string &n)
- void set_TeX_name (const std::string &n)
- std::string get_name () const
- std::string get_TeX_name () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*

- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.

- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.

- virtual ex eval_integ () const

  Evaluate integrals, if result is known.

- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.

- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.

- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.

- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).

- virtual size_t nops () const

  Number of operands/members.

- virtual ex op (size_t i) const

  Return operand/member at position i.

- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.

- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.

- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

- virtual void accept (GiNaC::visitor &v) const
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.

- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.

- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.

- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.

- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).

- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.

- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Additional Inherited Members

Protected Member Functions inherited from GiNaC::symbol

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for single differentiation of a symbol.
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes inherited from GiNaC::symbol

- unsigned serial

  unique serial number for comparison
- std::string name

  printname of this symbol
- std::string TeX_name

  LaTeX name of this symbol.

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.146.1 Detailed Description

Specialization of symbol to real domain.

8.146.2 Constructor & Destructor Documentation

8.146.2.1 realsymbol() [1/3]

```
GiNaC::realsymbol::realsymbol ()
```

References realsymbol(), and GiNaC::symbol::symbol().

Referenced by duplicate(), GiNaC::possymbol::possymbol(), GiNaC::possymbol::possymbol(), GiNaC::possymbol::possymbol(), and realsymbol().

8.146.2.2 realsymbol() [2/3]

```
GiNaC::realsymbol::realsymbol (
            const std::string & initname)  [explicit]
```

References GiNaC::symbol::symbol().

8.146.2.3 realsymbol() [3/3]

```
GiNaC::realsymbol::realsymbol (
            const std::string & initname,
            const std::string & texname)
```

References GiNaC::symbol::symbol().

8.146.3 Member Function Documentation

8.146.3.1 get_domain()

```
unsigned GiNaC::realsymbol::get_domain () const  [inline], [override], [virtual]
```

Reimplemented from GiNaC::symbol.

References GiNaC::domain::real.

8.146.3.2 conjugate()

```
ex GiNaC::realsymbol::conjugate () const  [inline], [override], [virtual]
```

Reimplemented from GiNaC::basic.

8.146.3.3 real_part()

```
ex GiNaC::realsymbol::real_part () const  [inline], [override], [virtual]
```

Reimplemented from GiNaC::basic.

8.146.3.4 imag_part()

```
ex GiNaC::realsymbol::imag_part () const  [inline], [override], [virtual]
```

Reimplemented from GiNaC::basic.

8.146.3.5 duplicate()

```
realsymbol * GiNaC::realsymbol::duplicate () const  [inline], [override], [virtual]
```

Create a clone of this object on the heap.

One can think of this as simulating a virtual copy constructor which is needed for instance by the refcounted construction of an ex from a basic.

Reimplemented from GiNaC::basic.

References GiNaC::status_flags::dynallocated, realsymbol(), and GiNaC::basic::setflag().

The documentation for this class was generated from the following files:

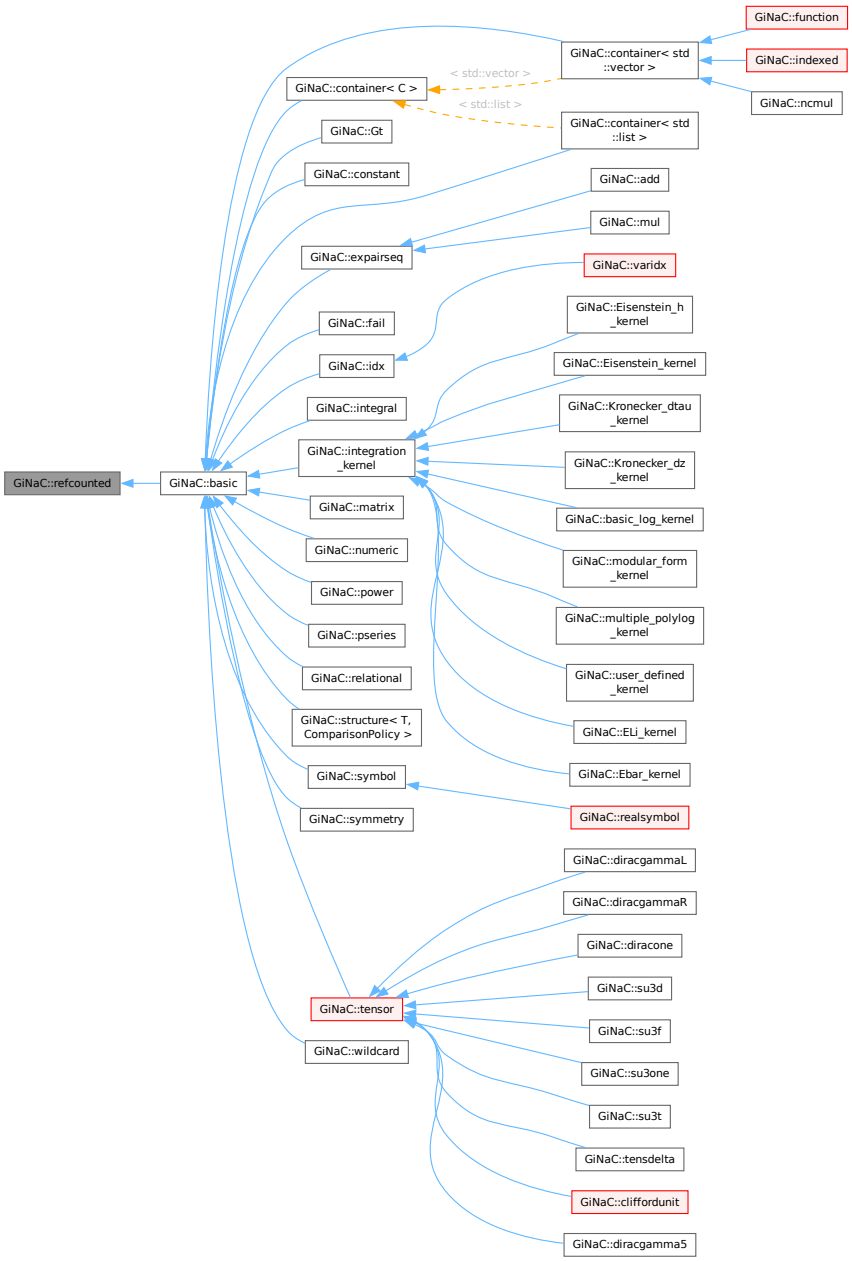
- symbol.h
- symbol.cpp

8.147 GiNaC::refcounted Class Reference

Base class for reference-counted objects.

```
#include <ptr.h>
```

Inheritance diagram for GiNaC::refcounted:



Public Member Functions

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Private Attributes

- unsigned int refcount

  reference counter

8.147.1 Detailed Description

Base class for reference-counted objects.

8.147.2 Constructor & Destructor Documentation

8.147.2.1 refcounted()

```
GiNaC::refcounted::refcounted ()  [inline], [noexcept]
```

References refcount.

8.147.3 Member Function Documentation

8.147.3.1 add_reference()

```
unsigned int GiNaC::refcounted::add_reference ()  [inline], [noexcept]
```

References refcount.

8.147.3.2 remove_reference()

```
unsigned int GiNaC::refcounted::remove_reference ()  [inline], [noexcept]
```

References refcount.

8.147.3.3 get_refcount()

```
unsigned int GiNaC::refcounted::get_refcount () const  [inline], [noexcept]
```

References refcount.

Referenced by GiNaC::ex::construct_from_basic(), GiNaC::basic::ensure_if_modifiable(), and GiNaC::basic::~basic().

8.147.3.4 set_refcount()

```
void GiNaC::refcounted::set_refcount (
            unsigned int r)  [inline], [noexcept]
```

References refcount.

Referenced by GiNaC::basic::operator=().

8.147.4 Member Data Documentation

8.147.4.1 refcount

```
unsigned int GiNaC::refcounted::refcount  [private]
```

reference counter

Referenced by add_reference(), get_refcount(), refcounted(), remove_reference(), and set_refcount().

The documentation for this class was generated from the following file:

- ptr.h

8.148 GiNaC::registered_class_options Class Reference

This class stores information about a registered GiNaC class.

```
#include <registrar.h>
```

Public Member Functions

- registered_class_options (const char *n, const char *p, const std::type_info &ti)
- const char * get_name () const
- const char * get_parent_name () const
- std::type_info const * get_id () const
- const std::vector< print_functor > & get_print_dispatch_table () const
- template<class Ctx, class T, class C>
  registered_class_options & print_func (void f(const T &, const C &c, unsigned))
- template<class Ctx, class T, class C>
  registered_class_options & print_func (void(T::*f)(const C &, unsigned))
- template<class Ctx>
  registered_class_options & print_func (const print_functor &f)
- void set_print_func (unsigned id, const print_functor &f)

Private Attributes

- const char * name
  Class name.
- const char * parent_name
  Name of superclass.
- std::type_info const * tinfo_key
  Type information key.
- std::vector< print_functor > print_dispatch_table
  Method table for print() dispatch.

8.148.1 Detailed Description

This class stores information about a registered GiNaC class.

8.148.2 Constructor & Destructor Documentation

8.148.2.1 registered_class_options()

```
GiNaC::registered_class_options::registered_class_options (
            const char * n,
            const char * p,
            const std::type_info & ti)  [inline]
```

References name, parent_name, and tinfo_key.

Referenced by print_func(), print_func(), and print_func().

8.148.3 Member Function Documentation

8.148.3.1 get_name()

```
const char * GiNaC::registered_class_options::get_name () const  [inline]
```

References name.

8.148.3.2 get_parent_name()

```
const char * GiNaC::registered_class_options::get_parent_name () const  [inline]
```

References parent_name.

8.148.3.3 get_id()

```
std::type_info const * GiNaC::registered_class_options::get_id () const  [inline]
```

References tinfo_key.

8.148.3.4 get_print_dispatch_table()

```
const std::vector< print_functor > & GiNaC::registered_class_options::get_print_dispatch_table
() const  [inline]
```

References print_dispatch_table.

Referenced by GiNaC::basic::print_dispatch().

8.148.3.5 print_func() [1/3]

```
template<class Ctx, class T, class C>
registered_class_options & GiNaC::registered_class_options::print_func (
            void  fconst T &, const C &c, unsigned)  [inline]
```

References registered_class_options(), and set_print_func().

8.148.3.6 print_func() [2/3]

```
template<class Ctx, class T, class C>
registered_class_options & GiNaC::registered_class_options::print_func (
            void(T::* f )(const C &, unsigned))  [inline]
```

References registered_class_options(), and set_print_func().

8.148.3.7 print_func() [3/3]

```
template<class Ctx>
registered_class_options & GiNaC::registered_class_options::print_func (
            const print_functor & f)  [inline]
```

References registered_class_options(), and set_print_func().

8.148.3.8 set_print_func()

```
void GiNaC::registered_class_options::set_print_func (
            unsigned id,
            const print_functor & f)  [inline]
```

References print_dispatch_table.

Referenced by print_func(), print_func(), and print_func().

8.148.4 Member Data Documentation

8.148.4.1 name

```
const char* GiNaC::registered_class_options::name  [private]
```

Class name.

Referenced by get_name(), and registered_class_options().

8.148.4.2 parent_name

```
const char* GiNaC::registered_class_options::parent_name  [private]
```

Name of superclass.

Referenced by get_parent_name(), and registered_class_options().

8.148.4.3 tinfo_key

```
std::type_info const* GiNaC::registered_class_options::tinfo_key  [private]
```

Type information key.

Referenced by get_id(), and registered_class_options().

8.148.4.4 print_dispatch_table

```
std::vector<print_functor> GiNaC::registered_class_options::print_dispatch_table  [private]
```

Method table for print() dispatch.

Referenced by get_print_dispatch_table(), and set_print_func().

The documentation for this class was generated from the following file:

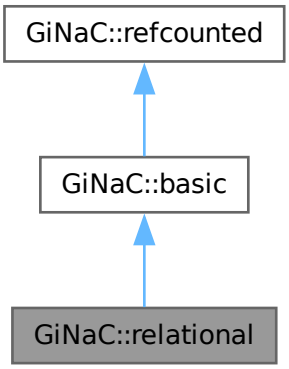
- registrar.h

8.149 GiNaC::relational Class Reference

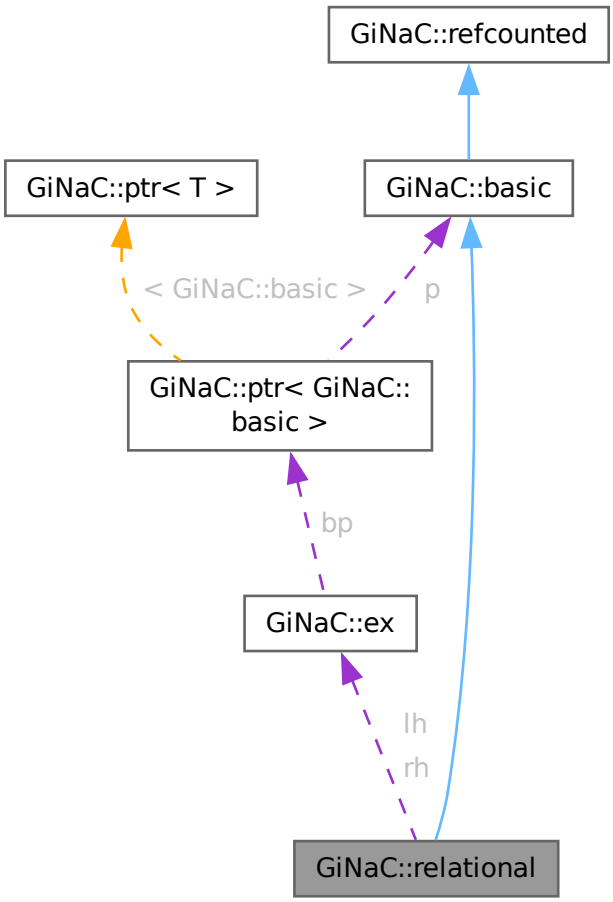
This class holds a relation consisting of two expressions and a logical relation between them.

```
#include <relational.h>
```

Inheritance diagram for GiNaC::relational:



Collaboration diagram for GiNaC::relational:



Classes

- struct safe_bool_helper

Public Types

- enum operators {
  equal , not_equal , less , less_or_equal ,
  greater , greater_or_equal }

Public Member Functions

- relational (const ex &lhs, const ex &rhs, operators oper=equal)
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

Information about the object.

- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex map (map_function &f) const override

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- ex canonical () const

  Returns an equivalent relational with zero right-hand side.
- ex lhs () const
- ex rhs () const
- operator safe_bool () const

  Cast the relational into a Boolean, mainly for evaluation within an if-statement.
- safe_bool operator! () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.

- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).

- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex eval_ncmul (const exvector &v) const override
- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void do_print (const print_context &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex lh
- ex rh
- operators o

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

Private Types

- typedef void(safe_bool_helper::*) safe_bool()

Private Member Functions

- safe_bool make_safe_bool (bool) const

8.149.1 Detailed Description

This class holds a relation consisting of two expressions and a logical relation between them.

8.149.2 Member Typedef Documentation

8.149.2.1 safe_bool

```
typedef void(safe_bool_helper::*) GiNaC::relational::safe_bool()  [private]
```

8.149.3 Member Enumeration Documentation

8.149.3.1 operators

```
enum GiNaC::relational::operators
```

Enumerator

equal	
not_equal	
less	
less_or_equal	

greater	
greater_or_equal	

8.149.4 Constructor & Destructor Documentation

8.149.4.1 relational()

```
GiNaC::relational::relational (
            const ex & lhs,
            const ex & rhs,
            operators oper = equal)
```

References GiNaC::basic::ex, lh, lhs(), o, rh, and rhs().

Referenced by GiNaC::pseries::add_series(), canonical(), GiNaC::pseries::derivative(), GiNaC::pseries::evalf(), GiNaC::pseries::expand(), match_same_type(), GiNaC::pseries::mul_const(), GiNaC::pseries::mul_series(), GiNaC::pseries::normal(), GiNaC::operator!=(), GiNaC::operator<(), GiNaC::operator<=(), GiNaC::operator==(), GiNaC::operator>(), GiNaC::operator>=(), GiNaC::pseries::power_const(), GiNaC::ex::series(), GiNaC::pseries::shift_exponents(), GiNaC::pseries::subs(), and subs().

8.149.5 Member Function Documentation

8.149.5.1 precedence()

```
unsigned GiNaC::relational::precedence () const  [inline], [override], [virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

Referenced by do_print().

8.149.5.2 info()

```
bool GiNaC::relational::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References equal, greater, greater_or_equal, less, less_or_equal, not_equal, o, GiNaC::info_flags::relation, GiNaC::info_flags::relation_equal, GiNaC::info_flags::relation_greater, GiNaC::info_flags::relation_greater_or_equal, GiNaC::info_flags::relation_less, GiNaC::info_flags::relation_less_or_equal, and GiNaC::info_flags::relation_not_equal.

Referenced by operator safe_bool().

8.149.5.3 nops()

```
size_t GiNaC::relational::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.149.5.4 op()

```
ex GiNaC::relational::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GINAC_ASSERT, lh, and rh.

8.149.5.5 map()

```
ex GiNaC::relational::map (
            map_function & f) const  [override], [virtual]
```

Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), GiNaC::dynallocate(), GiNaC::basic::ex, lh, o, and rh.

8.149.5.6 subs()

```
ex GiNaC::relational::subs (
            const exmap & m,
            unsigned options = 0) const  [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References GiNaC::are_ex_trivially_equal(), GiNaC::basic::ex, lh, o, relational(), rh, GiNaC::ex::subs(), and GiNaC::basic::subs_one_level().

8.149.5.7 archive()

```
void GiNaC::relational::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_ex(), GiNaC::archive_node::add_unsigned(), lh, o, and rh.

8.149.5.8 read_archive()

```
void GiNaC::relational::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::find_ex(), GiNaC::archive_node::find_unsigned(), lh, o, and rh.

8.149.5.9 canonical()

```
ex GiNaC::relational::canonical () const
```

Returns an equivalent relational with zero right-hand side.

References GiNaC::_ex0, GiNaC::basic::ex, lh, o, relational(), and rh.

8.149.5.10 eval_ncmul()

```
ex GiNaC::relational::eval_ncmul (
            const exvector & v) const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::ex::eval_ncmul(), GiNaC::basic::ex, and lh.

8.149.5.11 match_same_type()

```
bool GiNaC::relational::match_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic(), GINAC_ASSERT, GiNaC::is_exactly_a(), o, and relational().

8.149.5.12 return_type()

```
unsigned GiNaC::relational::return_type () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GINAC_ASSERT, lh, GiNaC::ex::return_type(), and rh.

8.149.5.13 return_type_tinfo()

```
return_type_t GiNaC::relational::return_type_tinfo () const  [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GINAC_ASSERT, lh, GiNaC::ex::return_type_tinfo(), and rh.

8.149.5.14 calchash()

```
unsigned GiNaC::relational::calchash () const  [override], [protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References equal, GiNaC::status_flags::evaluated, GiNaC::basic::flags, GiNaC::ex::gethash(), greater, greater_or_equal, GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, less, less_or_equal, lh, not_equal, o, rh, GiNaC::rotate_left(), and GiNaC::basic::setflag().

8.149.5.15 do_print()

```
void GiNaC::relational::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References lh, o, precedence(), GiNaC::ex::print(), GiNaC::print_operator(), rh, and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.149.5.16 do_print_python_repr()

```
void GiNaC::relational::do_print_python_repr (
             const print_python_repr & c,
             unsigned level) const  [protected]
```

References lh, o, GiNaC::ex::print(), GiNaC::print_operator(), rh, and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.149.5.17 lhs()

```
ex GiNaC::relational::lhs () const  [inline]
```

References GiNaC::basic::ex, and lh.

Referenced by GiNaC::atan_series(), GiNaC::atanh_series(), GiNaC::beta_series(), GiNaC::Li2_series(), GiNaC::log_series(), GiNaC::Order_series(), relational(), GiNaC::basic::series(), GiNaC::Eisenstein_h_kernel::series(), GiNaC::Eisenstein_kernel::series(), GiNaC::integration_kernel::series(), GiNaC::modular_form_kernel::series(), GiNaC::mul::series(), GiNaC::power::series(), GiNaC::pseries::series(), GiNaC::symbol::series(), GiNaC::tan_series(), and GiNaC::tanh_series().

8.149.5.18 rhs()

```
ex GiNaC::relational::rhs () const  [inline]
```

References GiNaC::basic::ex, and rh.

Referenced by GiNaC::atan_series(), GiNaC::atanh_series(), GiNaC::Li2_series(), GiNaC::log_series(), relational(), GiNaC::Eisenstein_h_kernel::series(), GiNaC::Eisenstein_kernel::series(), GiNaC::integration_kernel::series(), GiNaC::modular_form_kernel::series(), GiNaC::mul::series(), GiNaC::power::series(), GiNaC::pseries::series(), and GiNaC::symbol::series().

8.149.5.19 make_safe_bool()

```
relational::safe_bool GiNaC::relational::make_safe_bool (
            bool cond) const  [private]
```

References GiNaC::relational::safe_bool_helper::nonnull().

Referenced by operator safe_bool(), and operator!().

8.149.5.20 operator safe_bool()

```
GiNaC::relational::operator relational::safe_bool () const
```

Cast the relational into a Boolean, mainly for evaluation within an if-statement.

Note that (a<b) == false does not imply (a>=b) == true in the general symbolic case. A false result means the comparison is either false or undecidable (except of course for !=, where true means either unequal or undecidable).

References GiNaC::_num0_p, equal, GiNaC::basic::ex, GiNaC::ex_to(), greater, greater_or_equal, GiNaC::ex::info(), info(), GiNaC::is_exactly_a(), GiNaC::ex::is_zero(), GiNaC::is_zero(), less, less_or_equal, lh, make_safe_bool(), GiNaC::info_flags::negative, GiNaC::info_flags::nonnegative, not_equal, o, GiNaC::info_flags::positive, and rh.

8.149.5.21 operator"!()

```
relational::safe_bool GiNaC::relational::operator!  () const  [inline]
```

References make_safe_bool().

8.149.6 Member Data Documentation

8.149.6.1 lh

```
ex GiNaC::relational::lh  [protected]
```

Referenced by archive(), calchash(), canonical(), do_print(), do_print_python_repr(), eval_ncmul(), lhs(), map(), op(), operator safe_bool(), read_archive(), relational(), return_type(), return_type_tinfo(), and subs().

8.149.6.2 rh

```
ex GiNaC::relational::rh  [protected]
```

Referenced by archive(), calchash(), canonical(), do_print(), do_print_python_repr(), map(), op(), operator safe_bool(), read_archive(), relational(), return_type(), return_type_tinfo(), rhs(), and subs().

8.149.6.3 o

```
operators GiNaC::relational::o  [protected]
```

Referenced by archive(), calchash(), canonical(), do_print(), do_print_python_repr(), info(), map(), match_same_type(), operator safe_bool(), read_archive(), relational(), and subs().

The documentation for this class was generated from the following files:

- relational.h
- relational.cpp

8.150 GiNaC::remember_strategies Class Reference

Strategies how to clean up the function remember cache.

```
#include <flags.h>
```

Public Types

- enum { delete_never , delete_lru , delete_lfu , delete_cyclic }

8.150.1 Detailed Description

Strategies how to clean up the function remember cache.

See also

remember_table

8.150.2 Member Enumeration Documentation

8.150.2.1 anonymous enum

```
anonymous enum
```

Enumerator

delete_never	Let table grow undefinitely.
delete_lru	Least recently used.
delete_lfu	Least frequently used.
delete_cyclic	First (oldest) one in list.

The documentation for this class was generated from the following file:

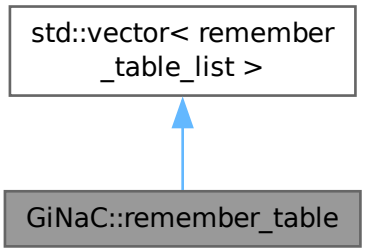
- flags.h

8.151 GiNaC::remember_table Class Reference

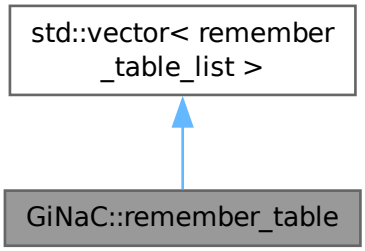
The remember table is organized like an n-fold associative cache in a microprocessor.

```
#include <remember.h>
```

Inheritance diagram for GiNaC::remember_table:



Collaboration diagram for GiNaC::remember_table:



Public Member Functions

- remember_table ()
- remember_table (unsigned s, unsigned as, unsigned strat)
- bool lookup_entry (function const &f, ex &result) const
- void add_entry (function const &f, ex const &result)
- void clear_all_entries ()
- void show_statistics (std::ostream &os, unsigned level) const

Static Public Member Functions

- static std::vector< remember_table > & remember_tables ()

Protected Member Functions

- void init_table ()

Protected Attributes

- unsigned table_size
- unsigned max_assoc_size
- unsigned remember_strategy

8.151.1 Detailed Description

The remember table is organized like an n-fold associative cache in a microprocessor.

The table has a width of 's' (which is rounded to table_size, some power of 2 near 's', internally) and a depth of 'as' (unless you choose that entries are never discarded). The place where an entry is stored depends on the hashvalue of the parameters of the function (this corresponds to the address of byte to be cached). The 'log_2(table_size)' least significant bits of this hashvalue give the slot in which the entry will be stored or looked up. Each slot can take up to 'as' entries. If a slot is full, an older entry is removed by one of the following strategies:

- oldest entry (the first one in the list)
- least recently used (the one with the lowest 'last_access')
- least frequently used (the one with the lowest 'successful_hits') or all entries are kept which means that the table grows indefinitely.

8.151.2 Constructor & Destructor Documentation

8.151.2.1 remember_table() [1/2]

```
GiNaC::remember_table::remember_table ()
```

References GiNaC::remember_strategies::delete_never, max_assoc_size, remember_strategy, and table_size.

Referenced by GiNaC::function::register_new().

8.151.2.2 remember_table() [2/2]

```
GiNaC::remember_table::remember_table (
            unsigned s,
            unsigned as,
            unsigned strat)
```

References init_table(), GiNaC::log2(), max_assoc_size, remember_strategy, and table_size.

8.151.3 Member Function Documentation

8.151.3.1 lookup_entry()

```
bool GiNaC::remember_table::lookup_entry (
            function const & f,
            ex & result) const
```

References GiNaC::basic::gethash(), GINAC_ASSERT, and table_size.

8.151.3.2 add_entry()

```
void GiNaC::remember_table::add_entry (
            function const & f,
            ex const & result)
```

References GiNaC::basic::gethash(), GINAC_ASSERT, and table_size.

8.151.3.3 clear_all_entries()

```
void GiNaC::remember_table::clear_all_entries ()
```

References init_table().

8.151.3.4 show_statistics()

```
void GiNaC::remember_table::show_statistics (
            std::ostream & os,
            unsigned level) const
```

References init_table(), and remember_tables().

8.151.3.5 remember_tables()

```
std::vector< remember_table > & GiNaC::remember_table::remember_tables ()  [static]
```

Referenced by GiNaC::function::lookup_remember_table(), GiNaC::function::register_new(), show_statistics(), and GiNaC::function::store_remember_table().

8.151.3.6 init_table()

```
void GiNaC::remember_table::init_table ()  [protected]
```

References max_assoc_size, remember_strategy, GiNaC::remember_table_list::remember_table_list(), and table_size.

Referenced by clear_all_entries(), remember_table(), and show_statistics().

8.151.4 Member Data Documentation

8.151.4.1 table_size

```
unsigned GiNaC::remember_table::table_size  [protected]
```

Referenced by add_entry(), init_table(), lookup_entry(), remember_table(), and remember_table().

8.151.4.2 max_assoc_size

```
unsigned GiNaC::remember_table::max_assoc_size  [protected]
```

Referenced by init_table(), remember_table(), and remember_table().

8.151.4.3 remember_strategy

```
unsigned GiNaC::remember_table::remember_strategy  [protected]
```

Referenced by init_table(), remember_table(), and remember_table().

The documentation for this class was generated from the following files:

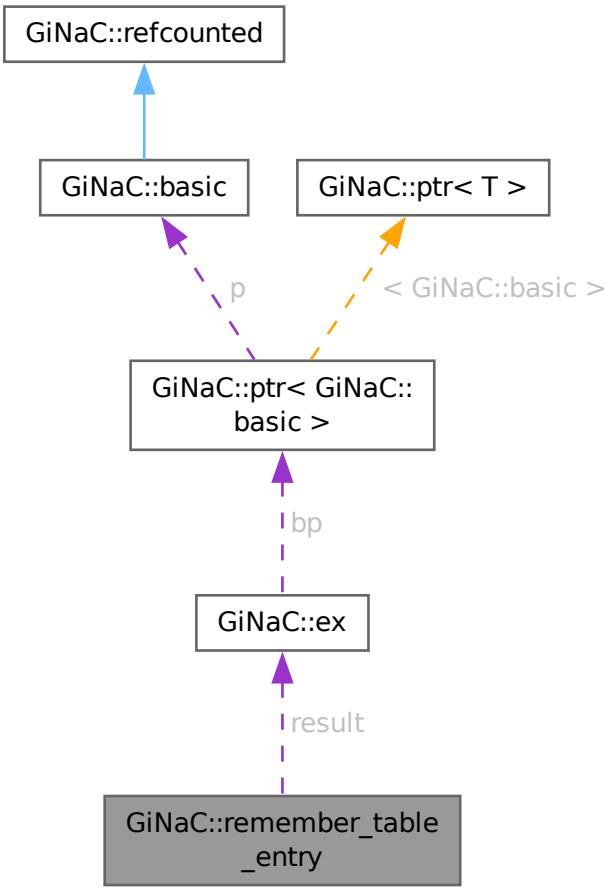
- remember.h
- remember.cpp

8.152 GiNaC::remember_table_entry Class Reference

A single entry in the remember table of a function.

`#include <remember.h>`

Collaboration diagram for GiNaC::remember_table_entry:



Public Member Functions

- remember_table_entry (function const &f, ex const &r)
- bool is_equal (function const &f) const
- ex get_result () const
- unsigned long get_last_access () const
- unsigned long get_successful_hits () const

Protected Attributes

- unsigned hashvalue
- exvector seq
- ex result
- unsigned long last_access
- unsigned successful_hits

Static Protected Attributes

- static unsigned long access_counter = 0

8.152.1 Detailed Description

A single entry in the remember table of a function.

Needs to be a friend of class function to access 'seq'. 'last_access' and 'successful_hits' are updated at each successful 'is_equal'.

8.152.2 Constructor & Destructor Documentation

8.152.2.1 remember_table_entry()

```
GiNaC::remember_table_entry::remember_table_entry (
            function const & f,
            ex const & r)
```

References access_counter, hashvalue, last_access, result, seq, and successful_hits.

Referenced by GiNaC::remember_table_list::add_entry().

8.152.3 Member Function Documentation

8.152.3.1 is_equal()

```
bool GiNaC::remember_table_entry::is_equal (
            function const & f) const
```

References access_counter, GiNaC::basic::gethash(), GINAC_ASSERT, hashvalue, is_equal(), last_access, GiNaC::container_storage< C >::seq, seq, and successful_hits.

Referenced by is_equal().

8.152.3.2 get_result()

```
ex GiNaC::remember_table_entry::get_result () const  [inline]
```

References result.

8.152.3.3 get_last_access()

```
unsigned long GiNaC::remember_table_entry::get_last_access () const  [inline]
```

References last_access.

8.152.3.4 get_successful_hits()

```
unsigned long GiNaC::remember_table_entry::get_successful_hits () const  [inline]
```

References successful_hits.

8.152.4 Member Data Documentation

8.152.4.1 hashvalue

```
unsigned GiNaC::remember_table_entry::hashvalue  [protected]
```

Referenced by is_equal(), and remember_table_entry().

8.152.4.2 seq

```
exvector GiNaC::remember_table_entry::seq  [protected]
```

Referenced by is_equal(), and remember_table_entry().

8.152.4.3 result

```
ex GiNaC::remember_table_entry::result  [protected]
```

Referenced by get_result(), and remember_table_entry().

8.152.4.4 last_access

```
unsigned long GiNaC::remember_table_entry::last_access  [mutable], [protected]
```

Referenced by get_last_access(), is_equal(), and remember_table_entry().

8.152.4.5 successful_hits

```
unsigned GiNaC::remember_table_entry::successful_hits  [mutable], [protected]
```

Referenced by get_successful_hits(), is_equal(), and remember_table_entry().

8.152.4.6 access_counter

```
unsigned long GiNaC::remember_table_entry::access_counter = 0  [static], [protected]
```

Referenced by is_equal(), and remember_table_entry().

The documentation for this class was generated from the following files:

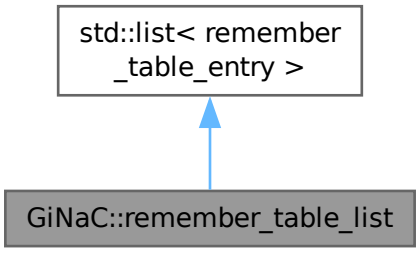
- remember.h
- remember.cpp

8.153 GiNaC::remember_table_list Class Reference

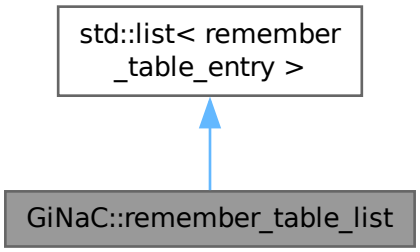
A list of entries in the remember table having some least significant bits of the hashvalue in common.

`#include <remember.h>`

Inheritance diagram for GiNaC::remember_table_list:



Collaboration diagram for GiNaC::remember_table_list:



Public Member Functions

- remember_table_list (unsigned as, unsigned strat)
- void add_entry (function const &f, ex const &result)
- bool lookup_entry (function const &f, ex &result) const

Protected Attributes

- unsigned max_assoc_size
- unsigned remember_strategy

8.153.1 Detailed Description

A list of entries in the remember table having some least significant bits of the hashvalue in common.

8.153.2 Constructor & Destructor Documentation

8.153.2.1 remember_table_list()

```
GiNaC::remember_table_list::remember_table_list (
            unsigned as,
            unsigned strat)
```

References max_assoc_size, and remember_strategy.

Referenced by GiNaC::remember_table::init_table().

8.153.3 Member Function Documentation

8.153.3.1 add_entry()

```
void GiNaC::remember_table_list::add_entry (
            function const & f,
            ex const & result)
```

References GiNaC::remember_strategies::delete_cyclic, GiNaC::remember_strategies::delete_lfu, GiNaC::remember_strategies::dele GiNaC::remember_strategies::delete_never, GINAC_ASSERT, max_assoc_size, remember_strategy, and GiNaC::remember_table_entry::remember_table_entry().

8.153.3.2 lookup_entry()

```
bool GiNaC::remember_table_list::lookup_entry (
            function const & f,
            ex & result) const
```

8.153.4 Member Data Documentation

8.153.4.1 max_assoc_size

```
unsigned GiNaC::remember_table_list::max_assoc_size  [protected]
```

Referenced by add_entry(), and remember_table_list().

8.153.4.2 remember_strategy

```
unsigned GiNaC::remember_table_list::remember_strategy  [protected]
```

Referenced by add_entry(), and remember_table_list().

The documentation for this class was generated from the following files:

- remember.h
- remember.cpp

8.154 GiNaC::return_type_t Struct Reference

To distinguish between different kinds of non-commutative objects.

```
#include <registrar.h>
```

Public Member Functions

- bool operator< (const return_type_t &other) const

  Strict weak ordering (so one can put return_type_t's into a STL container).
- bool operator== (const return_type_t &other) const
- bool operator!= (const return_type_t &other) const

Public Attributes

- std::type_info const * tinfo

  to distinguish between non-commutative objects of different type.
- unsigned rl

  to distinguish between non-commutative objects of the same type.

8.154.1 Detailed Description

To distinguish between different kinds of non-commutative objects.

8.154.2 Member Function Documentation

8.154.2.1 operator<()

```
bool GiNaC::return_type_t::operator< (
            const return_type_t & other) const  [inline]
```

Strict weak ordering (so one can put return_type_t's into a STL container).

References rl, and tinfo.

8.154.2.2 operator==()

```
bool GiNaC::return_type_t::operator== (
            const return_type_t & other) const  [inline]
```

References rl, and tinfo.

Referenced by operator!=().

8.154.2.3 operator"!=()

```
bool GiNaC::return_type_t::operator!= (
            const return_type_t & other) const  [inline]
```

References operator==().

8.154.3 Member Data Documentation

8.154.3.1 tinfo

```
std::type_info const* GiNaC::return_type_t::tinfo
```

to distinguish between non-commutative objects of different type.

Referenced by GiNaC::is_clifford_tinfo(), GiNaC::is_color_tinfo(), GiNaC::make_return_type_t(), operator<(), operator==(), GiNaC::basic::return_type_tinfo(), and GiNaC::structure< T, ComparisonPolicy >::return_type_tinfo().

8.154.3.2 rl

```
unsigned GiNaC::return_type_t::rl
```

to distinguish between non-commutative objects of the same type.

Think of gamma matrices with different representation labels.

Referenced by GiNaC::get_representation_label(), GiNaC::make_return_type_t(), operator<(), operator==(), GiNaC::basic::return_type_tinfo(), and GiNaC::structure< T, ComparisonPolicy >::return_type_tinfo().

The documentation for this struct was generated from the following file:

- registrar.h

8.155 GiNaC::return_types Class Reference

```
#include <flags.h>
```

Public Types

- enum { commutative , noncommutative , noncommutative_composite }

8.155.1 Member Enumeration Documentation

8.155.1.1 anonymous enum

```
anonymous enum
```

Enumerator

commutative	
noncommutative	
noncommutative_composite	

The documentation for this class was generated from the following file:

- flags.h

8.156 GiNaC::relational::safe_bool_helper Struct Reference

Public Member Functions

- void nonnull ()

8.156.1 Member Function Documentation

8.156.1.1 nonnull()

```
void GiNaC::relational::safe_bool_helper::nonnull ()  [inline]
```

Referenced by GiNaC::relational::make_safe_bool().

The documentation for this struct was generated from the following file:

- relational.h

8.157 GiNaC::scalar_products Class Reference

Helper class for storing information about known scalar products which are to be automatically replaced by simplify_indexed().

```
#include <indexed.h>
```

Public Member Functions

- void add (const ex &v1, const ex &v2, const ex &sp)
  Register scalar product pair and its value.
- void add (const ex &v1, const ex &v2, const ex &dim, const ex &sp)
  Register scalar product pair and its value for a specific space dimension.
- void add_vectors (const lst &l, const ex &dim=wild())
  Register list of vectors.
- void clear ()
  Clear all registered scalar products.
- bool is_defined (const ex &v1, const ex &v2, const ex &dim) const
  Check whether scalar product pair is defined.
- ex evaluate (const ex &v1, const ex &v2, const ex &dim) const
  Return value of defined scalar product pair.
- void debugprint () const

Protected Attributes

- spmap spm

8.157.1 Detailed Description

Helper class for storing information about known scalar products which are to be automatically replaced by simplify_indexed().

See also

simplify_indexed

8.157.2 Member Function Documentation

8.157.2.1 add() [1/2]

```
void GiNaC::scalar_products::add (
             const ex & v1,
             const ex & v2,
             const ex & sp)
```

Register scalar product pair and its value.

References spm, and GiNaC::spmapkey::spmapkey().

8.157.2.2 add() [2/2]

```
void GiNaC::scalar_products::add (
             const ex & v1,
             const ex & v2,
             const ex & dim,
             const ex & sp)
```

Register scalar product pair and its value for a specific space dimension.

References spm, and GiNaC::spmapkey::spmapkey().

8.157.2.3 add_vectors()

```
void GiNaC::scalar_products::add_vectors (
             const lst & l,
             const ex & dim = wild())
```

Register list of vectors.

This adds all possible pairs of products a.i * b.i with the value a*b (note that this is not a scalar vector product but an ordinary product of scalars).

References GiNaC::add::add().

8.157.2.4 clear()

```
void GiNaC::scalar_products::clear ()
```

Clear all registered scalar products.

References spm.

8.157.2.5 is_defined()

```
bool GiNaC::scalar_products::is_defined (
             const ex & v1,
             const ex & v2,
             const ex & dim) const
```

Check whether scalar product pair is defined.

References spm, and GiNaC::spmapkey::spmapkey().

Referenced by GiNaC::indexed::simplify_indexed_product.

8.157.2.6 evaluate()

```
ex GiNaC::scalar_products::evaluate (
             const ex & v1,
             const ex & v2,
             const ex & dim) const
```

Return value of defined scalar product pair.

References spm, and GiNaC::spmapkey::spmapkey().

Referenced by GiNaC::indexed::simplify_indexed_product.

8.157.2.7 debugprint()

```
void GiNaC::scalar_products::debugprint () const
```

References GiNaC::spmapkey::debugprint(), and spm.

8.157.3 Member Data Documentation

8.157.3.1 spm

```
spmap GiNaC::scalar_products::spm  [protected]
```

Referenced by add(), add(), clear(), debugprint(), evaluate(), and is_defined().

The documentation for this class was generated from the following files:

- indexed.h
- indexed.cpp

8.158 GiNaC::series_options Class Reference

Flags to control series expansion.

```
#include <flags.h>
```

Public Types

- enum { suppress_branchcut = 0x0001 }

8.158.1 Detailed Description

Flags to control series expansion.

8.158.2 Member Enumeration Documentation

8.158.2.1 anonymous enum

```
anonymous enum
```

Enumerator

suppress_branchcut	Suppress branch cuts in series expansion. Branch cuts manifest themselves as step functions, if this option is not passed. If it is passed and expansion at a point on a cut is performed, then the analytic continuation of the function is expanded.

The documentation for this class was generated from the following file:

- flags.h

8.159 GiNaC::solve_algo Class Reference

Switch to control algorithm for linear system solving.

`#include <flags.h>`

Public Types

- enum {
  automatic , gauss , divfree , bareiss ,
  markowitz }

8.159.1 Detailed Description

Switch to control algorithm for linear system solving.

8.159.2 Member Enumeration Documentation

8.159.2.1 anonymous enum

`anonymous enum`

Enumerator

automatic	Let the system choose. A heuristics is applied for automatic determination of a suitable algorithm.
gauss	Gauss elimination. If $m_{i,j}^{(0)}$ are the entries of the original matrix, then the matrix is transformed into triangular form by applying the rules $$m_{i,j}^{(k+1)} = m_{i,j}^{(k)} - m_{i,k}^{(k)} m_{k,j}^{(k)} / m_{k,k}^{(k)}$$ This algorithm is well-suited for numerical matrices but generally suffers from the expensive division (and computation of GCDs) at each step.
divfree	Division-free elimination. This is a modification of Gauss elimination where the division by the pivot element is not carried out. If $m_{i,j}^{(0)}$ are the entries of the original matrix, then the matrix is transformed into triangular form by applying the rules $$m_{i,j}^{(k+1)} = m_{i,j}^{(k)} m_{k,k}^{(k)} - m_{i,k}^{(k)} m_{k,j}^{(k)}$$ This algorithm is only there for the purpose of cross-checks. It suffers from exponential intermediate expression swell. Use it only for small systems.

bareiss	Bareiss fraction-free elimination. This is a modification of Gauss elimination where the division by the pivot element is *delayed* until it can be carried out without computing GCDs. If $m_{i,j}^{(0)}$ are the entries of the original matrix, then the matrix is transformed into triangular form by applying the rules $$m_{i,j}^{(k+1)} = (m_{i,j}^{(k)} m_{k,k}^{(k)} - m_{i,k}^{(k)} m_{k,j}^{(k)}) / m_{k-1,k-1}^{(k-1)}$$ (We have set $m_{-1,-1}^{(-1)} = 1$ in order to avoid a case distinction in above formula.) It can be shown that nothing more than polynomial long division is needed for carrying out the division. This is generally the fastest algorithm for solving linear systems. In contrast to division-free elimination it only has a linear expression swell. For two-dimensional systems, the two algorithms are equivalent, however.
markowitz	Markowitz-ordered Gaussian elimination. Same as the usual Gaussian elimination, but with additional effort spent on selecting pivots that minimize fill-in. Faster than the methods above for large sparse matrices (particularly with symbolic coefficients), otherwise slightly slower than Gaussian elimination.

The documentation for this class was generated from the following file:

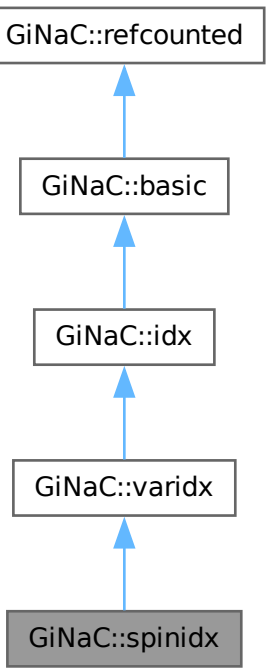
- flags.h

8.160 GiNaC::spinidx Class Reference

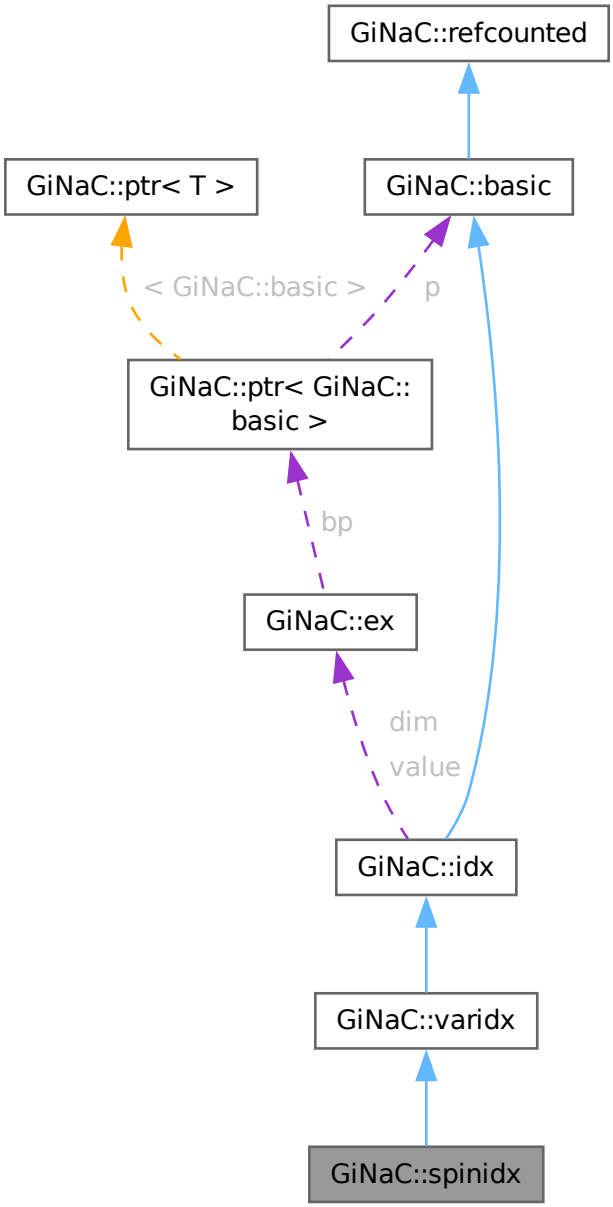
This class holds a spinor index that can be dotted or undotted and that also has a variance.

`#include <idx.h>`

Inheritance diagram for GiNaC::spinidx:



Collaboration diagram for GiNaC::spinidx:



Public Member Functions

- spinidx (const ex &v, const ex &dim=2, bool covariant=false, bool dotted=false)

  Construct index with given value, dimension, variance and dot.
- bool is_dummy_pair_same_type (const basic &other) const override

  Check whether the index forms a dummy index pair with another index of the same type.
- ex conjugate () const override
- void archive (archive_node &n) const override

Save (serialize) the object into archive node.

- void read_archive (const archive_node &n, lst &syms) override

  Load (deserialize) the object from an archive node.
- bool is_dotted () const

  Check whether the index is dotted.
- bool is_undotted () const

  Check whether the index is not dotted.
- ex toggle_dot () const

  Make a new index with the same value and variance but the opposite dottedness.
- ex toggle_variance_dot () const

  Make a new index with the same value but opposite variance and dottedness.

Public Member Functions inherited from GiNaC::varidx

- varidx (const ex &v, const ex &dim, bool covariant=false)

  Construct index with given value, dimension and variance.
- bool is_dummy_pair_same_type (const basic &other) const override

  Check whether the index forms a dummy index pair with another index of the same type.
- void archive (archive_node &n) const override

  Save (serialize) the object into archive node.
- void read_archive (const archive_node &n, lst &syms) override

  Load (deserialize) the object from an archive node.
- bool is_covariant () const

  Check whether the index is covariant.
- bool is_contravariant () const

  Check whether the index is contravariant (not covariant).
- ex toggle_variance () const

  Make a new index with the same value but the opposite variance.

Public Member Functions inherited from GiNaC::idx

- idx (const ex &v, const ex &dim)

  Construct index with given value and dimension.
- bool info (unsigned inf) const override

  Information about the object.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex map (map_function &f) const override

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- ex evalf () const override

  By default, basic::evalf would evaluate the index value but we don't want a.1 to become a.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex get_value () const

  Get value of index.
- bool is_numeric () const

  Check whether the index is numeric.

- bool is_symbolic () const

  Check whether the index is symbolic.
- ex get_dim () const

  Get dimension of index space.
- bool is_dim_numeric () const

  Check whether the dimension is numeric.
- bool is_dim_symbolic () const

  Check whether the dimension is symbolic.
- ex replace_dim (const ex &new_dim) const

  Make a new index with the same value but a different dimension.
- ex minimal_dim (const idx &other) const

  Return the minimum of the dimensions of this and another index.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

Check whether this is a polynomial in the given variables.

- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const

- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const

Protected Member Functions inherited from GiNaC::varidx

- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const

Protected Member Functions inherited from GiNaC::idx

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for an index always returns 0.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void print_index (const print_context &c, unsigned level) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_csrc (const print_csrc &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- bool dotted

Protected Attributes inherited from GiNaC::varidx

- bool covariant

  x.mu, default is contravariant: x∼mu

Protected Attributes inherited from GiNaC::idx

- ex value

  Expression that constitutes the index (numeric or symbolic name).
- ex dim

  Dimension of space (can be symbolic or numeric).

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.160.1 Detailed Description

This class holds a spinor index that can be dotted or undotted and that also has a variance.

This is used in the Weyl-van-der-Waerden formalism where the dot indicates complex conjugation. There is an associated (asymmetric) metric tensor that can be used to raise/lower spinor indices.

8.160.2 Constructor & Destructor Documentation

8.160.2.1 spinidx()

```
GiNaC::spinidx::spinidx (
            const ex & v,
            const ex & dim = 2,
            bool covariant = false,
            bool dotted = false)
```

Construct index with given value, dimension, variance and dot.

Parameters

v	Value of index (numeric or symbolic)
dim	Dimension of index space (numeric or symbolic)

covariant	Make covariant index (default is contravariant)
dotted	Make covariant dotted (default is undotted)

References dotted.

Referenced by is_dummy_pair_same_type(), match_same_type(), toggle_dot(), and toggle_variance_dot().

8.160.3 Member Function Documentation

8.160.3.1 is_dummy_pair_same_type()

```
bool GiNaC::spinidx::is_dummy_pair_same_type (
             const basic & other) const  [override], [virtual]
```

Check whether the index forms a dummy index pair with another index of the same type.

Reimplemented from GiNaC::idx.

References dotted, and spinidx().

8.160.3.2 conjugate()

```
ex GiNaC::spinidx::conjugate () const  [inline], [override], [virtual]
```

Reimplemented from GiNaC::basic.

References toggle_dot().

8.160.3.3 archive()

```
void GiNaC::spinidx::archive (
             archive_node & n) const  [override], [virtual]
```

Save (serialize) the object into archive node.

Archive the object.

Losely speaking, this method turns an expression into a byte stream (which can be saved and restored later on, or sent via network, etc.)

Reimplemented from GiNaC::idx.

References GiNaC::archive_node::add_bool(), and dotted.

8.160.3.4 read_archive()

```
void GiNaC::spinidx::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented from GiNaC::idx.

References dotted, and GiNaC::archive_node::find_bool().

8.160.3.5 match_same_type()

```
bool GiNaC::spinidx::match_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::idx.

References dotted, GINAC_ASSERT, GiNaC::is_a(), and spinidx().

8.160.3.6 is_dotted()

```
bool GiNaC::spinidx::is_dotted () const  [inline]
```

Check whether the index is dotted.

References dotted.

8.160.3.7 is_undotted()

```
bool GiNaC::spinidx::is_undotted () const  [inline]
```

Check whether the index is not dotted.

References dotted.

8.160.3.8 toggle_dot()

```
ex GiNaC::spinidx::toggle_dot () const
```

Make a new index with the same value and variance but the opposite dottedness.

References GiNaC::basic::clearflag(), dotted, GiNaC::basic::duplicate(), GiNaC::status_flags::hash_calculated, and spinidx().

Referenced by conjugate().

8.160.3.9 toggle_variance_dot()

```
ex GiNaC::spinidx::toggle_variance_dot () const
```

Make a new index with the same value but opposite variance and dottedness.

References GiNaC::basic::clearflag(), GiNaC::varidx::covariant, dotted, GiNaC::basic::duplicate(), GiNaC::status_flags::hash_calculat and spinidx().

8.160.3.10 do_print()

```
void GiNaC::spinidx::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References GiNaC::varidx::covariant, dotted, GiNaC::idx::print_index(), and GiNaC::print_context::s.

8.160.3.11 do_print_latex()

```
void GiNaC::spinidx::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

References dotted, GiNaC::idx::print_index(), and GiNaC::print_context::s.

8.160.3.12 do_print_tree()

```
void GiNaC::spinidx::do_print_tree (
             const print_tree & c,
             unsigned level) const  [protected]
```

References GiNaC::varidx::covariant, GiNaC::print_tree::delta_indent, GiNaC::idx::dim, dotted, GiNaC::basic::flags, GiNaC::basic::hashvalue, GiNaC::ex::print(), GiNaC::print_context::s, and GiNaC::idx::value.

8.160.4 Member Data Documentation

8.160.4.1 dotted

```
bool GiNaC::spinidx::dotted  [protected]
```

Referenced by archive(), do_print(), do_print_latex(), do_print_tree(), is_dotted(), is_dummy_pair_same_type(), is_undotted(), match_same_type(), read_archive(), spinidx(), toggle_dot(), and toggle_variance_dot().

The documentation for this class was generated from the following files:

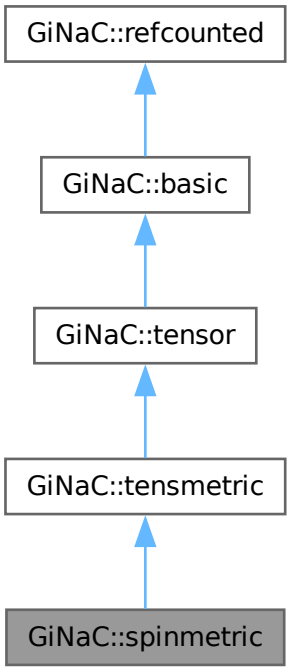
- idx.h
- idx.cpp

8.161 GiNaC::spinmetric Class Reference

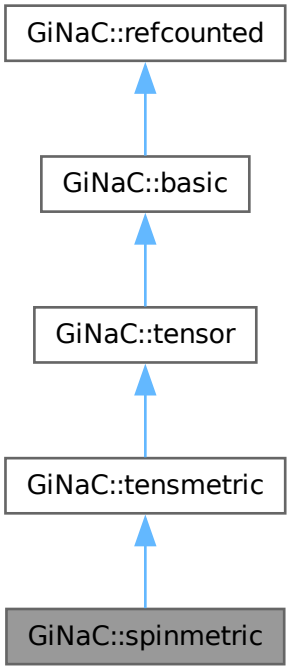
This class represents an antisymmetric spinor metric tensor which can be used to raise/lower indices of 2-component Weyl spinors.

```
#include <tensor.h>
```

Inheritance diagram for GiNaC::spinmetric:



Collaboration diagram for GiNaC::spinmetric:



Public Member Functions

- bool info (unsigned inf) const override

  Information about the object.
- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of an indexed metric tensor.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of an indexed spinor metric with something else.

Public Member Functions inherited from GiNaC::tensmetric

- bool info (unsigned inf) const override

  Information about the object.
- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of an indexed metric tensor.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of an indexed metric tensor with something else.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ∼basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

Return coefficient of degree n in object s.

- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensmetric

- unsigned return_type () const override
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.161.1 Detailed Description

This class represents an antisymmetric spinor metric tensor which can be used to raise/lower indices of 2-component Weyl spinors.

If indexed, it must have exactly two indices of the same type which must be of class spinidx or a subclass and have dimension 2.

8.161.2 Member Function Documentation

8.161.2.1 info()

```
bool GiNaC::spinmetric::info (
             unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References GiNaC::info_flags::real.

8.161.2.2 eval_indexed()

```
ex GiNaC::spinmetric::eval_indexed (
             const basic & i) const  [override], [virtual]
```

Automatic symbolic evaluation of an indexed metric tensor.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::_ex_1, GiNaC::ex_to(), GiNaC::idx::get_value(), GINAC_ASSERT, GiNaC::basic::hold(), GiNaC::is_a(), GiNaC::info_flags::nonnegint, GiNaC::basic::nops(), and GiNaC::basic::op().

8.161.2.3 contract_with()

```
bool GiNaC::spinmetric::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [override], [virtual]
```

Contraction of an indexed spinor metric with something else.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::_ex2, GiNaC::_ex_2, GiNaC::delta_tensor(), GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_a(), GiNaC::varidx::is_covariant(), GiNaC::is_dummy_pair(), and GiNaC::idx::is_symbolic().

8.161.2.4 do_print()

```
void GiNaC::spinmetric::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

8.161.2.5 do_print_latex()

```
void GiNaC::spinmetric::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

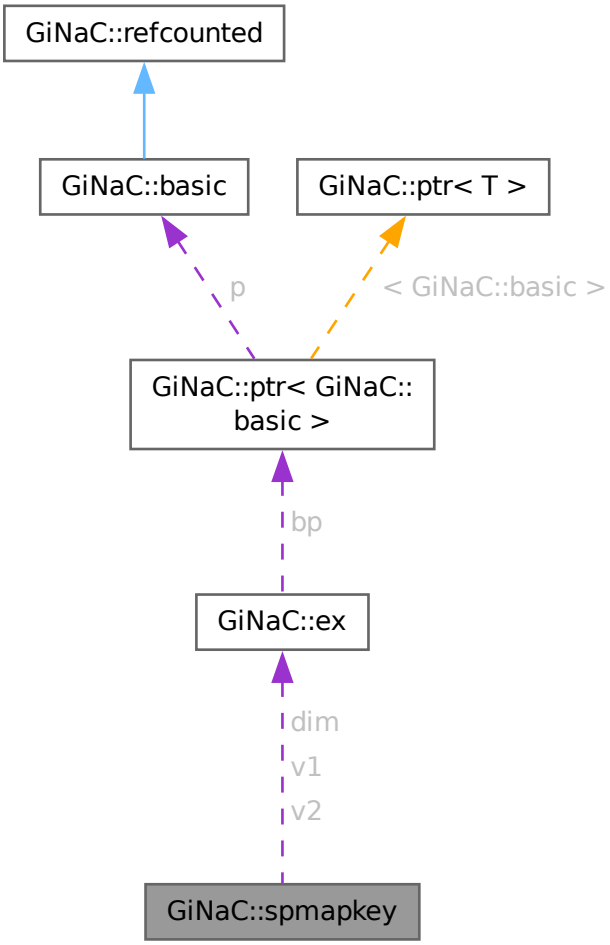
The documentation for this class was generated from the following files:

- tensor.h
- tensor.cpp

8.162 GiNaC::spmapkey Class Reference

```
#include <indexed.h>
```

Collaboration diagram for GiNaC::spmapkey:



Public Member Functions

- spmapkey ()
- spmapkey (const ex &v1, const ex &v2, const ex &dim=wild())
- bool operator== (const spmapkey &other) const
- bool operator< (const spmapkey &other) const
- void debugprint () const

Protected Attributes

- ex v1
- ex v2
- ex dim

8.162.1 Constructor & Destructor Documentation

8.162.1.1 spmapkey() [1/2]

```
GiNaC::spmapkey::spmapkey ()  [inline]
```

References dim, and GiNaC::wild().

Referenced by GiNaC::scalar_products::add(), GiNaC::scalar_products::add(), GiNaC::scalar_products::evaluate(), GiNaC::scalar_products::is_defined(), operator<(), and operator==().

8.162.1.2 spmapkey() [2/2]

```
GiNaC::spmapkey::spmapkey (
             const ex & v1,
             const ex & v2,
             const ex & dim = wild())
```

References GiNaC::ex::compare(), dim, GiNaC::is_a(), GiNaC::ex::op(), v1, and v2.

8.162.2 Member Function Documentation

8.162.2.1 operator==()

```
bool GiNaC::spmapkey::operator== (
             const spmapkey & other) const
```

References dim, GiNaC::is_a(), GiNaC::ex::is_equal(), spmapkey(), v1, and v2.

8.162.2.2 operator<()

```
bool GiNaC::spmapkey::operator< (
             const spmapkey & other) const
```

References GiNaC::ex::compare(), dim, GiNaC::is_a(), spmapkey(), v1, and v2.

8.162.2.3 debugprint()

```
void GiNaC::spmapkey::debugprint () const
```

References dim, v1, and v2.

Referenced by GiNaC::scalar_products::debugprint().

8.162.3 Member Data Documentation

8.162.3.1 v1

```
ex GiNaC::spmapkey::v1  [protected]
```

Referenced by debugprint(), operator<(), operator==(), and spmapkey().

8.162.3.2 v2

```
ex GiNaC::spmapkey::v2  [protected]
```

Referenced by debugprint(), operator<(), operator==(), and spmapkey().

8.162.3.3 dim

```
ex GiNaC::spmapkey::dim  [protected]
```

Referenced by debugprint(), operator<(), operator==(), spmapkey(), and spmapkey().

The documentation for this class was generated from the following files:

- indexed.h
- indexed.cpp

8.163 GiNaC::status_flags Class Reference

Flags to store information about the state of an object.

```
#include <flags.h>
```

Public Types

- enum {
  dynallocated = 0x0001 , evaluated = 0x0002 , expanded = 0x0004 , hash_calculated = 0x0008 ,
  not_shareable = 0x0010 , has_indices = 0x0020 , has_no_indices = 0x0040 , is_positive = 0x0080 ,
  is_negative = 0x0100 , purely_indefinite = 0x0200 }

8.163.1 Detailed Description

Flags to store information about the state of an object.

See also

basic::flags

8.163.2 Member Enumeration Documentation

8.163.2.1 anonymous enum

```
anonymous enum
```

Enumerator

dynallocated	heap-allocated (i.e. created by new if we want to be clever and bypass the stack, **See also** ex::construct_from_basic())
evaluated	.eval() has already done its job
expanded	.expand(0) has already done its job (other expand() options ignore this flag)
hash_calculated	.calchash() has already done its job
not_shareable	don't share instances of this object between different expressions unless explicitly asked to (used by ex::compare())
has_indices	
has_no_indices	
is_positive	
is_negative	
purely_indefinite	

The documentation for this class was generated from the following file:

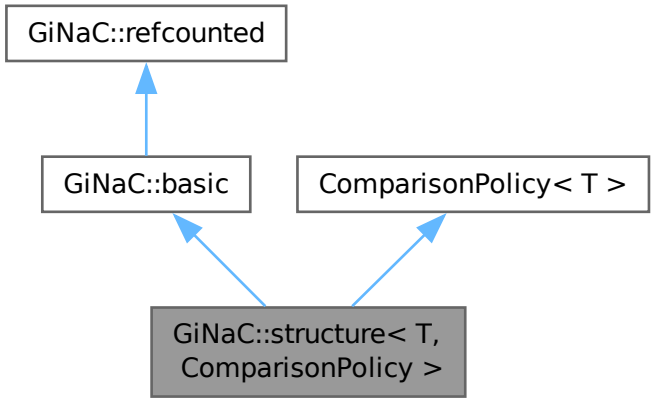
- flags.h

8.164 GiNaC::structure< T, ComparisonPolicy > Class Template Reference

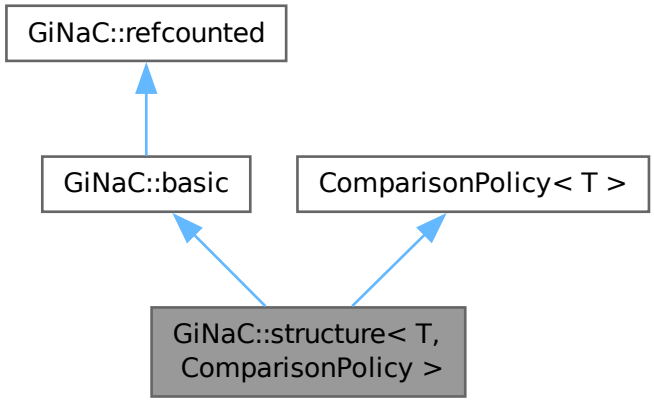
Wrapper template for making GiNaC classes out of C++ structures.

```
#include <structure.h>
```

Inheritance diagram for GiNaC::structure< T, ComparisonPolicy >:



Collaboration diagram for GiNaC::structure< T, ComparisonPolicy >:



Public Member Functions

- structure (const T &t)

  Construct structure as a copy of a given C++ structure.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex evalm () const override

  Evaluate sums, products and integer powers of matrices.
- ex eval_indexed (const basic &i) const override

Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

- void print (const print_context &c, unsigned level=0) const override

  Output to stream.
- unsigned precedence () const override

  Return relative operator precedence (for parenthezing output).
- bool info (unsigned inf) const override

  Information about the object.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex operator[ ] (const ex &index) const override
- ex operator[ ] (size_t i) const override
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- ex & operator[ ] (const ex &index) override
- ex & operator[ ] (size_t i) override
- bool has (const ex &other, unsigned options=0) const override

  Test for occurrence of a pattern.
- bool match (const ex &pattern, exmap &repl_lst) const override

  Check whether the expression matches a given pattern.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex map (map_function &f) const override

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- int degree (const ex &s) const override

  Return degree of highest power in object s.
- int ldegree (const ex &s) const override

  Return degree of lowest power in object s.
- ex coeff (const ex &s, int n=1) const override

  Return coefficient of degree n in object s.
- ex expand (unsigned options=0) const override

  Expand expression, i.e.
- ex collect (const ex &s, bool distributed=false) const override

  Sort expanded expression in terms of powers of some object(s).
- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const override

  Default implementation of ex::normal().
- ex to_rational (exmap &repl) const override

  Default implementation of ex::to_rational().
- ex to_polynomial (exmap &repl) const override
- numeric integer_content () const override
- ex smod (const numeric &xi) const override

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- numeric max_coefficient () const override

  Implementation ex::max_coefficient().
- exvector get_free_indices () const override

  Return a vector containing the free indices of an expression.
- ex add_indexed (const ex &self, const ex &other) const override

  Add two indexed expressions.

- ex scalar_mul_indexed (const ex &self, const numeric &other) const override

  Multiply an indexed expression with a scalar.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Try to contract two indexed expressions that appear in the same product.
- unsigned return_type () const override
- return_type_t return_type_tinfo () const override
- const T * operator-> () const
- T & get_struct ()
- const T & get_struct () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

Stop further evaluation.

- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex eval_ncmul (const exvector &v) const override
- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- ex derivative (const symbol &s) const override

  Default implementation of ex::diff().
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Static Private Member Functions

- static const char * get_class_name ()

Private Attributes

- T obj

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.164.1 Detailed Description

template<class T, template< class > class ComparisonPolicy>
class GiNaC::structure< T, ComparisonPolicy >

Wrapper template for making GiNaC classes out of C++ structures.

8.164.2 Constructor & Destructor Documentation

8.164.2.1 structure()

```
template<class T, template< class > class ComparisonPolicy>
GiNaC::structure< T, CP >::structure (
            const T & t)  [inline]
```

Construct structure as a copy of a given C++ structure.

Default constructor.

References obj.

Referenced by is_equal_same_type().

8.164.3 Member Function Documentation

8.164.3.1 get_class_name()

```
template<class T, template< class > class ComparisonPolicy>
const char * GiNaC::structure< T, ComparisonPolicy >::get_class_name ()  [inline], [static],
[private]
```

References get_class_name().

Referenced by get_class_name().

8.164.3.2 eval()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::eval () const  [inline], [override], [virtual]
```

Perform automatic non-interruptive term rewriting rules.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, and GiNaC::basic::hold().

8.164.3.3 evalm()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::evalm () const  [inline], [override], [virtual]
```

Evaluate sums, products and integer powers of matrices.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.4 eval_ncmul()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::eval_ncmul (
            const exvector & v) const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, and GiNaC::hold_ncmul().

8.164.3.5 eval_indexed()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::eval_indexed (
            const basic & i) const  [inline], [override], [virtual]
```

Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic(), GiNaC::basic::ex, and GiNaC::basic::hold().

8.164.3.6 print()

```
template<class T, template< class > class ComparisonPolicy>
void GiNaC::structure< T, ComparisonPolicy >::print (
             const print_context & c,
             unsigned level = 0) const  [inline], [override], [virtual]
```

Output to stream.

This performs double dispatch on the dynamic type of ∗this and the dynamic type of the supplied print context.

Parameters

c	print context object that describes the output formatting
level	value that is used to identify the precedence or indentation level for placing parentheses and formatting

Reimplemented from GiNaC::basic.

8.164.3.7 precedence()

```
template<class T, template< class > class ComparisonPolicy>
unsigned GiNaC::structure< T, ComparisonPolicy >::precedence () const  [inline], [override],
[virtual]
```

Return relative operator precedence (for parenthezing output).

Reimplemented from GiNaC::basic.

8.164.3.8 info()

```
template<class T, template< class > class ComparisonPolicy>
bool GiNaC::structure< T, ComparisonPolicy >::info (
             unsigned inf) const  [inline], [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

8.164.3.9 nops()

```
template<class T, template< class > class ComparisonPolicy>
size_t GiNaC::structure< T, ComparisonPolicy >::nops () const  [inline], [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.164.3.10 op()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::op (
            size_t i) const  [inline], [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.11 operator[]() [1/4]

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::operator[] (
            const ex & index) const  [inline], [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.12 operator[]() [2/4]

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::operator[] (
            size_t i) const  [inline], [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.13 let_op()

```
template<class T, template< class > class ComparisonPolicy>
ex & GiNaC::structure< T, ComparisonPolicy >::let_op (
            size_t i)  [inline], [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.14 operator[]() [3/4]

```
template<class T, template< class > class ComparisonPolicy>
ex & GiNaC::structure< T, ComparisonPolicy >::operator[] (
            const ex & index)  [inline], [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.15 operator[]() [4/4]

```
template<class T, template< class > class ComparisonPolicy>
ex & GiNaC::structure< T, ComparisonPolicy >::operator[] (
             size_t i)  [inline], [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.16 has()

```
template<class T, template< class > class ComparisonPolicy>
bool GiNaC::structure< T, ComparisonPolicy >::has (
             const ex & pattern,
             unsigned options = 0) const  [inline], [override], [virtual]
```

Test for occurrence of a pattern.

An object 'has' a pattern if it matches the pattern itself or one of the children 'has' it. As a consequence (according to the definition of children) given e=x+y+z, e.has(x) is true but e.has(x+y) is false.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.17 match()

```
template<class T, template< class > class ComparisonPolicy>
bool GiNaC::structure< T, ComparisonPolicy >::match (
             const ex & pattern,
             exmap & repl_lst) const  [inline], [override], [virtual]
```

Check whether the expression matches a given pattern.

For every wildcard object in the pattern, a pair with the wildcard as a key and matching expression as a value is added to repl_lst.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.18 match_same_type()

```
template<class T, template< class > class ComparisonPolicy>
bool GiNaC::structure< T, ComparisonPolicy >::match_same_type (
             const basic & other) const  [inline], [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic().

8.164.3.19 subs()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::subs (
            const exmap & m,
            unsigned options = 0) const  [inline], [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.20 map()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::map (
            map_function & f) const  [inline], [override], [virtual]
```

Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.21 degree()

```
template<class T, template< class > class ComparisonPolicy>
int GiNaC::structure< T, ComparisonPolicy >::degree (
            const ex & s) const  [inline], [override], [virtual]
```

Return degree of highest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.22 ldegree()

```
template<class T, template< class > class ComparisonPolicy>
int GiNaC::structure< T, ComparisonPolicy >::ldegree (
            const ex & s) const  [inline], [override], [virtual]
```

Return degree of lowest power in object s.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.23 coeff()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::coeff (
             const ex & s,
             int n = 1) const  [inline], [override], [virtual]
```

Return coefficient of degree n in object s.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.24 expand()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::expand (
             unsigned options = 0) const  [inline], [override], [virtual]
```

Expand expression, i.e.

multiply it out and return the result as a new expression.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.25 collect()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::collect (
             const ex & s,
             bool distributed = false) const  [inline], [override], [virtual]
```

Sort expanded expression in terms of powers of some object(s).

Parameters

s	object(s) to sort in
distributed	recursive or distributed form (only used when s is a list)

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.26 derivative()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::derivative (
            const symbol & s) const  [inline], [override], [protected], [virtual]
```

Default implementation of ex::diff().

It maps the operation on the operands (or returns 0 when the object has no operands).

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.27 series()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::series (
            const relational & r,
            int order,
            unsigned options = 0) const  [inline], [override], [virtual]
```

Default implementation of ex::series().

This performs Taylor expansion.

See also

ex::series

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.28 normal()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::normal (
            exmap & repl,
            exmap & rev_lookup,
            lst & modifier) const  [inline], [override], [virtual]
```

Default implementation of ex::normal().

It normalizes the children and replaces the object with a temporary symbol.

See also

ex::normal

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.29 to_rational()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::to_rational (
            exmap & repl) const  [inline], [override], [virtual]
```

Default implementation of ex::to_rational().

This replaces the object with a temporary symbol.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.30 to_polynomial()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::to_polynomial (
            exmap & repl) const  [inline], [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.31 integer_content()

```
template<class T, template< class > class ComparisonPolicy>
numeric GiNaC::structure< T, ComparisonPolicy >::integer_content () const  [inline], [override],
[virtual]
```

Reimplemented from GiNaC::basic.

8.164.3.32 smod()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::smod (
            const numeric & xi) const  [inline], [override], [virtual]
```

Apply symmetric modular homomorphism to an expanded multivariate polynomial.

This function is usually used internally by heur_gcd().

Parameters

xi	modulus

Returns

mapped polynomial

See also

heur_gcd

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.33 max_coefficient()

```
template<class T, template< class > class ComparisonPolicy>
numeric GiNaC::structure< T, ComparisonPolicy >::max_coefficient () const  [inline], [override],
[virtual]
```

Implementation ex::max_coefficient().

See also

heur_gcd

Reimplemented from GiNaC::basic.

8.164.3.34 get_free_indices()

```
template<class T, template< class > class ComparisonPolicy>
exvector GiNaC::structure< T, ComparisonPolicy >::get_free_indices () const  [inline], [override],
[virtual]
```

Return a vector containing the free indices of an expression.

Reimplemented from GiNaC::basic.

8.164.3.35 add_indexed()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::add_indexed (
            const ex & self,
            const ex & other) const  [inline], [override], [virtual]
```

Add two indexed expressions.

They are guaranteed to be of class indexed (or a subclass) and their indices are compatible. This function is used internally by simplify_indexed().

Parameters

self	First indexed expression; its base object is ∗this
other	Second indexed expression

Returns

sum of self and other

See also

ex::simplify_indexed()

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.164.3.36 scalar_mul_indexed()

```
template<class T, template< class > class ComparisonPolicy>
ex GiNaC::structure< T, ComparisonPolicy >::scalar_mul_indexed (
            const ex & self,
            const numeric & other) const  [inline], [override], [virtual]
```

Multiply an indexed expression with a scalar.

This function is used internally by simplify_indexed().

Parameters

self	Indexed expression; its base object is *this
other	Numeric value

Returns

product of self and other

See also

ex::simplify_indexed()

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, and GiNaC::ex::ex().

8.164.3.37 contract_with()

```
template<class T, template< class > class ComparisonPolicy>
bool GiNaC::structure< T, ComparisonPolicy >::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [inline], [override], [virtual]
```

Try to contract two indexed expressions that appear in the same product.

If a contraction exists, the function overwrites one or both of the expressions and returns true. Otherwise it returns false. It is guaranteed that both expressions are of class indexed (or a subclass) and that at least one dummy index has been found. This functions is used internally by simplify_indexed().

Parameters

self	Pointer to first indexed expression; its base object is *this
other	Pointer to second indexed expression
v	The complete vector of factors

Returns

true if the contraction was successful, false otherwise

See also

ex::simplify_indexed()

Reimplemented from GiNaC::basic.

8.164.3.38 return_type()

```
template<class T, template< class > class ComparisonPolicy>
unsigned GiNaC::structure< T, ComparisonPolicy >::return_type () const  [inline], [override],
[virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative.

8.164.3.39 return_type_tinfo()

```
template<class T, template< class > class ComparisonPolicy>
return_type_t GiNaC::structure< T, ComparisonPolicy >::return_type_tinfo () const  [inline],
[override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_type_t::rl, and GiNaC::return_type_t::tinfo.

8.164.3.40 is_equal_same_type()

```
template<class T, template< class > class ComparisonPolicy>
bool GiNaC::structure< T, ComparisonPolicy >::is_equal_same_type (
            const basic & other) const  [inline], [override], [protected], [virtual]
```

Returns true if two objects of same type are equal.

Normally needs not be reimplemented as long as it wasn't overwritten by some parent class, since it just calls compare_same_type(). The reason why this function exists is that sometimes it is easier to determine equality than an order relation and then it can be overridden.

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic(), GINAC_ASSERT, GiNaC::is_a(), obj, and structure().

8.164.3.41 calchash()

```
template<class T, template< class > class ComparisonPolicy>
unsigned GiNaC::structure< T, ComparisonPolicy >::calchash () const  [inline], [override],
[protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

8.164.3.42 operator->()

```
template<class T, template< class > class ComparisonPolicy>
const T * GiNaC::structure< T, ComparisonPolicy >::operator-> () const  [inline]
```

References obj.

8.164.3.43 get_struct() [1/2]

```
template<class T, template< class > class ComparisonPolicy>
T & GiNaC::structure< T, ComparisonPolicy >::get_struct ()  [inline]
```

References obj.

8.164.3.44 get_struct() [2/2]

```
template<class T, template< class > class ComparisonPolicy>
const T & GiNaC::structure< T, ComparisonPolicy >::get_struct () const  [inline]
```

References obj.

8.164.4 Member Data Documentation

8.164.4.1 obj

```
template<class T, template< class > class ComparisonPolicy>
T GiNaC::structure< T, ComparisonPolicy >::obj  [private]
```

Referenced by get_struct(), get_struct(), is_equal_same_type(), operator->(), and structure().

The documentation for this class was generated from the following file:

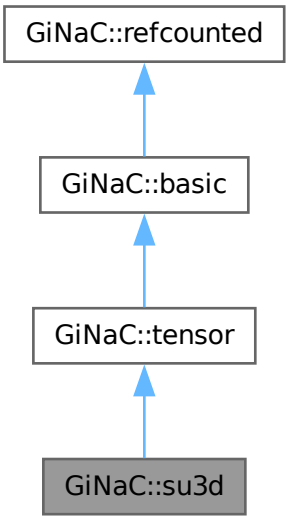
- structure.h

8.165 GiNaC::su3d Class Reference

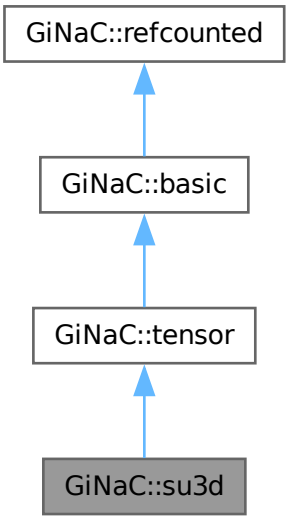
This class represents the tensor of symmetric su(3) structure constants.

`#include <color.h>`

Inheritance diagram for GiNaC::su3d:



Collaboration diagram for GiNaC::su3d:



Public Member Functions

- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of indexed symmetric structure constant.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of an indexed symmetric structure constant with something else.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)

- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

Load (deserialize) the object from an archive node.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- unsigned return_type () const override
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.165.1 Detailed Description

This class represents the tensor of symmetric su(3) structure constants.

8.165.2 Member Function Documentation

8.165.2.1 eval_indexed()

```
ex GiNaC::su3d::eval_indexed (
            const basic & i) const  [override], [virtual]
```

Automatic symbolic evaluation of indexed symmetric structure constant.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1_2, GiNaC::_ex1_3, GiNaC::_ex3, GiNaC::_ex_1_2, GiNaC::_ex_1_3, GiNaC::_ex_6, CMPINDICES, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::basic::hold(), GiNaC::is_a(), GiNaC::info_flags::nonnegint, GiNaC::basic::nops(), GiNaC::basic::op(), GiNaC::sqrt(), and std::swap().

8.165.2.2 contract_with()

```
bool GiNaC::su3d::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [override], [virtual]
```

Contraction of an indexed symmetric structure constant with something else.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::color_T(), GiNaC::delta_tensor(), GiNaC::ex_to(), GiNaC::find_free_and_dummy(), GiNaC::get_representation_label(), GINAC_ASSERT, GiNaC::is_a(), GiNaC::is_exactly_a(), GiNaC::numeric::numeric(), and GiNaC::permute_free_index_to_front().

8.165.2.3 return_type()

```
unsigned GiNaC::su3d::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative.

8.165.2.4 do_print()

```
void GiNaC::su3d::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

8.165.2.5 do_print_latex()

```
void GiNaC::su3d::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

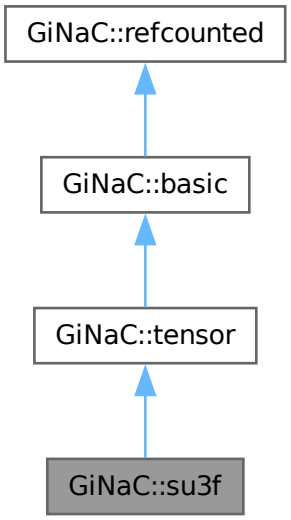
- color.h
- color.cpp

8.166 GiNaC::su3f Class Reference

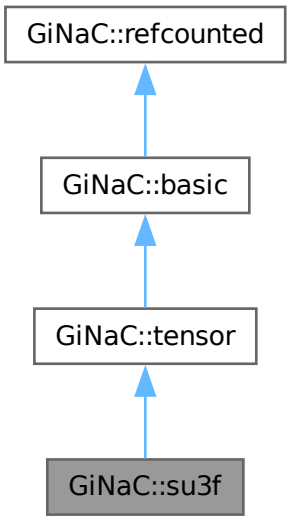
This class represents the tensor of antisymmetric su(3) structure constants.

```
#include <color.h>
```

Inheritance diagram for GiNaC::su3f:



Collaboration diagram for GiNaC::su3f:



Public Member Functions

- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of indexed antisymmetric structure constant.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of an indexed antisymmetric structure constant with something else.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.

- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().

- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- unsigned return_type () const override
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.166.1 Detailed Description

This class represents the tensor of antisymmetric su(3) structure constants.

8.166.2 Member Function Documentation

8.166.2.1 eval_indexed()

```
ex GiNaC::su3f::eval_indexed (
            const basic & i) const  [override], [virtual]
```

Automatic symbolic evaluation of indexed antisymmetric structure constant.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1_2, GiNaC::_ex3, GiNaC::_ex_1_2, CMPINDICES, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::basic::hold(), GiNaC::is_a(), GiNaC::info_flags::nonnegint, GiNaC::basic::nops(), GiNaC::basic::op(), GiNaC::sqrt(), and std::swap().

8.166.2.2 contract_with()

```
bool GiNaC::su3f::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [override], [virtual]
```

Contraction of an indexed antisymmetric structure constant with something else.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::color_T(), GiNaC::delta_tensor(), GiNaC::ex_to(), GiNaC::get_representation_label(), GINAC_ASSERT, GiNaC::I, GiNaC::is_a(), GiNaC::is_exactly_a(), GiNaC::numeric::numeric(), and GiNaC::permute_free_index_to_fr

8.166.2.3 return_type()

```
unsigned GiNaC::su3f::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative.

8.166.2.4 do_print()

```
void GiNaC::su3f::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

8.166.2.5 do_print_latex()

```
void GiNaC::su3f::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

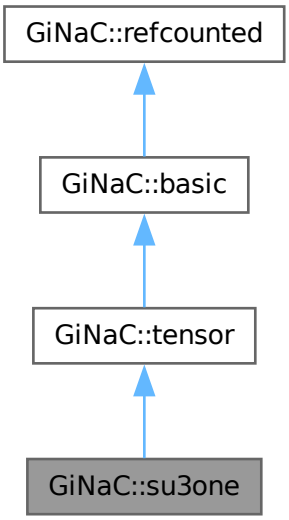
- color.h
- color.cpp

8.167 GiNaC::su3one Class Reference

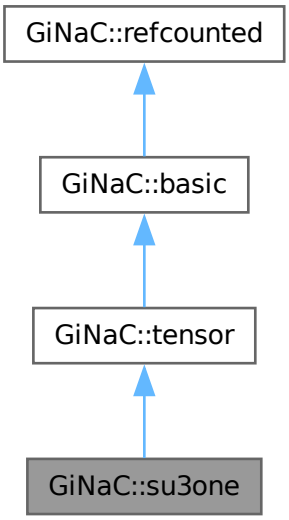
This class represents the su(3) unity element.

`#include <color.h>`

Inheritance diagram for GiNaC::su3one:



Collaboration diagram for GiNaC::su3one:



Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*

- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.

- virtual basic * duplicate () const

  Create a clone of this object on the heap.

- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.

- virtual ex evalf () const

  Evaluate object numerically.

- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.

- virtual ex eval_integ () const

  Evaluate integrals, if result is known.

- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.

- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.

- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.

- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).

- virtual bool info (unsigned inf) const

  Information about the object.

- virtual size_t nops () const

  Number of operands/members.

- virtual ex op (size_t i) const

  Return operand/member at position i.

- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.

- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.

- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.

- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).

- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.

- virtual int degree (const ex &s) const

Return degree of highest power in object s.

- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

Stop further evaluation.

- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.167.1 Detailed Description

This class represents the su(3) unity element.

8.167.2 Member Function Documentation

8.167.2.1 do_print()

```
void GiNaC::su3one::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

8.167.2.2 do_print_latex()

```
void GiNaC::su3one::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

The documentation for this class was generated from the following file:

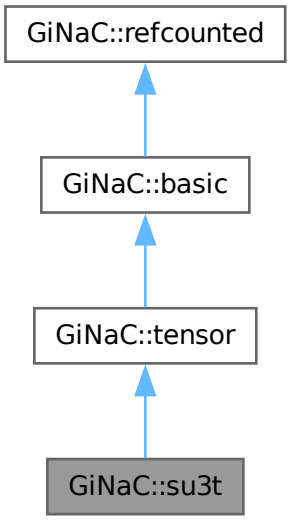
- color.h

8.168 GiNaC::su3t Class Reference

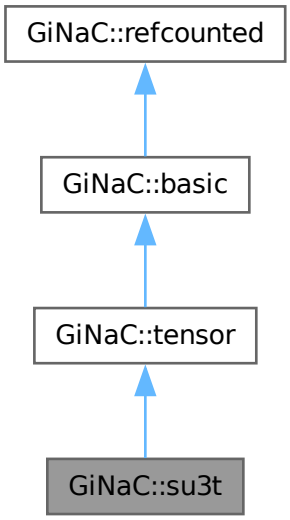
This class represents an su(3) generator.

`#include <color.h>`

Inheritance diagram for GiNaC::su3t:



Collaboration diagram for GiNaC::su3t:



Public Member Functions

- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of generator with something else.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)

- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

Load (deserialize) the object from an archive node.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

Ensure the object may be modified without hurting others, throws if this is not the case.

- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.168.1 Detailed Description

This class represents an su(3) generator.

8.168.2 Member Function Documentation

8.168.2.1 contract_with()

```
bool GiNaC::su3t::contract_with (
             exvector::iterator self,
             exvector::iterator other,
             exvector & v) const  [override], [virtual]
```

Contraction of generator with something else.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::color_ONE(), GiNaC::color_trace(), GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_a(), GiNaC::is_exactly_a(), and GiNaC::numeric::numeric().

8.168.2.2 do_print()

```
void GiNaC::su3t::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

8.168.2.3 do_print_latex()

```
void GiNaC::su3t::do_print_latex (
            const print_latex & c,
            unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

- color.h
- color.cpp

8.169 GiNaC::subs_options Class Reference

Flags to control the behavior of subs().

`#include <flags.h>`

Public Types

- enum {
  no_pattern = 0x0001 , subs_no_pattern = 0x0001 , algebraic = 0x0002 , subs_algebraic = 0x0002 ,
  pattern_is_product = 0x0004 , pattern_is_not_product = 0x0008 , no_index_renaming = 0x0010 ,
  really_subs_idx = 0x0020 }

8.169.1 Detailed Description

Flags to control the behavior of subs().

8.169.2 Member Enumeration Documentation

8.169.2.1 anonymous enum

```
anonymous enum
```

Enumerator

no_pattern	disable pattern matching
subs_no_pattern	
algebraic	enable algebraic substitutions
subs_algebraic	
pattern_is_product	used internally by expairseq::subschildren()
pattern_is_not_product	used internally by expairseq::subschildren()
no_index_renaming	
really_subs_idx	

The documentation for this class was generated from the following file:

- flags.h

8.170 GiNaC::sy_is_less Class Reference

Public Member Functions

- sy_is_less (exvector::iterator v_)
- bool operator() (const ex &lh, const ex &rh) const

Private Attributes

- exvector::iterator v

8.170.1 Constructor & Destructor Documentation

8.170.1.1 sy_is_less()

```
GiNaC::sy_is_less::sy_is_less (
            exvector::iterator v_)  [inline]
```

References v.

Referenced by GiNaC::symmetry::canonicalize.

8.170.2 Member Function Documentation

8.170.2.1 operator()()

```
bool GiNaC::sy_is_less::operator() (
            const ex & lh,
            const ex & rh) const  [inline]
```

References GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), and v.

8.170.3 Member Data Documentation

8.170.3.1 v

```
exvector::iterator GiNaC::sy_is_less::v  [private]
```

Referenced by operator()(), and sy_is_less().

The documentation for this class was generated from the following file:

- symmetry.cpp

8.171 GiNaC::sy_swap Class Reference

Public Member Functions

- sy_swap (exvector::iterator v_, bool &s)
- void operator() (const ex &lh, const ex &rh)

Public Attributes

- bool & swapped

Private Attributes

- exvector::iterator v

8.171.1 Constructor & Destructor Documentation

8.171.1.1 sy_swap()

```
GiNaC::sy_swap::sy_swap (
            exvector::iterator v_,
            bool & s)  [inline]
```

References swapped, and v.

Referenced by GiNaC::symmetry::canonicalize.

8.171.2 Member Function Documentation

8.171.2.1 operator()()

```
void GiNaC::sy_swap::operator() (
            const ex & lh,
            const ex & rh)  [inline]
```

References GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), swapped, and v.

8.171.3 Member Data Documentation

8.171.3.1 v

```
exvector::iterator GiNaC::sy_swap::v  [private]
```

Referenced by operator()(), and sy_swap().

8.171.3.2 swapped

```
bool& GiNaC::sy_swap::swapped
```

Referenced by operator()(), and sy_swap().

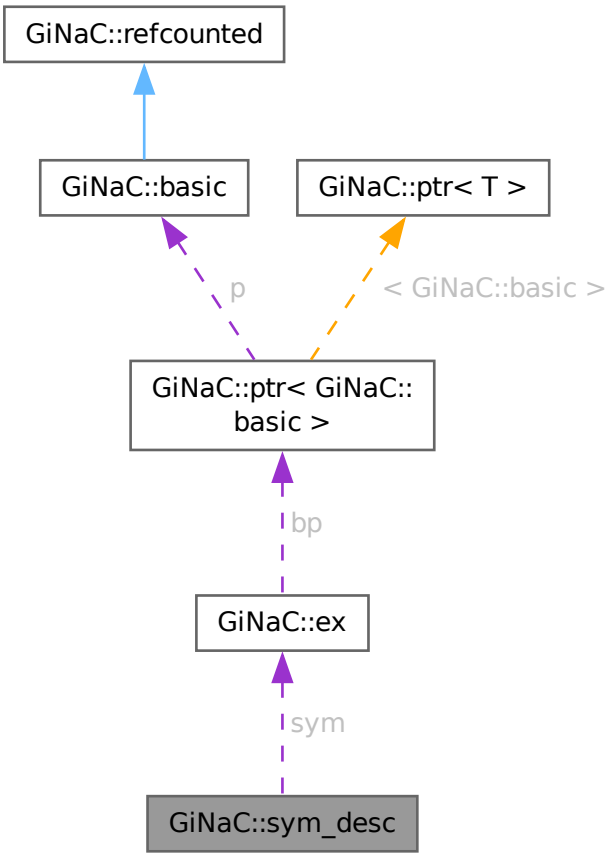
The documentation for this class was generated from the following file:

- symmetry.cpp

8.172 GiNaC::sym_desc Struct Reference

This structure holds information about the highest and lowest degrees in which a symbol appears in two multivariate polynomials "a" and "b".

Collaboration diagram for GiNaC::sym_desc:



Public Member Functions

- sym_desc (const ex &s)

  Initialize symbol, leave other variables uninitialized.
- bool operator< (const sym_desc &x) const

  Comparison operator for sorting.

Public Attributes

- ex sym

  Reference to symbol.
- int deg_a

  Highest degree of symbol in polynomial "a".
- int deg_b

  Highest degree of symbol in polynomial "b".
- int ldeg_a

  Lowest degree of symbol in polynomial "a".
- int ldeg_b

  Lowest degree of symbol in polynomial "b".
- int max_deg

  Maximum of deg_a and deg_b (Used for sorting).
- size_t max_lcnops

  Maximum number of terms of leading coefficient of symbol in both polynomials.

8.172.1 Detailed Description

This structure holds information about the highest and lowest degrees in which a symbol appears in two multivariate polynomials "a" and "b".

A vector of these structures with information about all symbols in two polynomials can be created with the function get_symbol_stats().

See also

get_symbol_stats

8.172.2 Constructor & Destructor Documentation

8.172.2.1 sym_desc()

```
GiNaC::sym_desc::sym_desc (
            const ex & s)  [inline]
```

Initialize symbol, leave other variables uninitialized.

References deg_a, deg_b, ldeg_a, ldeg_b, max_deg, max_lcnops, and sym.

Referenced by GiNaC::add_symbol(), and operator<().

8.172.3 Member Function Documentation

8.172.3.1 operator<()

```
bool GiNaC::sym_desc::operator< (
            const sym_desc & x) const  [inline]
```

Comparison operator for sorting.

References max_deg, max_lcnops, and sym_desc().

8.172.4 Member Data Documentation

8.172.4.1 sym

```
ex GiNaC::sym_desc::sym
```

Reference to symbol.

Referenced by sym_desc().

8.172.4.2 deg_a

```
int GiNaC::sym_desc::deg_a
```

Highest degree of symbol in polynomial "a".

Referenced by sym_desc().

8.172.4.3 deg_b

```
int GiNaC::sym_desc::deg_b
```

Highest degree of symbol in polynomial "b".

Referenced by sym_desc().

8.172.4.4 ldeg_a

```
int GiNaC::sym_desc::ldeg_a
```

Lowest degree of symbol in polynomial "a".

Referenced by sym_desc().

8.172.4.5 ldeg_b

```
int GiNaC::sym_desc::ldeg_b
```

Lowest degree of symbol in polynomial "b".

Referenced by sym_desc().

8.172.4.6 max_deg

```
int GiNaC::sym_desc::max_deg
```

Maximum of deg_a and deg_b (Used for sorting).

Referenced by operator<(), and sym_desc().

8.172.4.7 max_lcnops

```
size_t GiNaC::sym_desc::max_lcnops
```

Maximum number of terms of leading coefficient of symbol in both polynomials.

Referenced by operator<(), and sym_desc().

The documentation for this struct was generated from the following file:

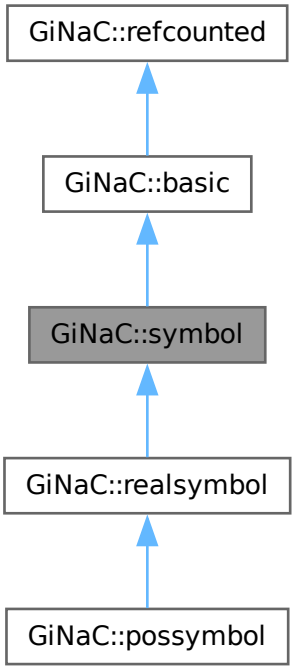
- normal.cpp

8.173 GiNaC::symbol Class Reference

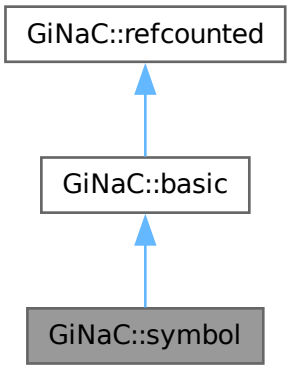
Basic CAS symbol.

```
#include <symbol.h>
```

Inheritance diagram for GiNaC::symbol:



Collaboration diagram for GiNaC::symbol:



Public Member Functions

- symbol (const std::string &initname)

- symbol (const std::string &initname, const std::string &texname)
- bool info (unsigned inf) const override

  Information about the object.
- ex eval () const override

  Perform automatic non-interruptive term rewriting rules.
- ex evalf () const override

  Evaluate object numerically.
- ex series (const relational &s, int order, unsigned options=0) const override

  Implementation of ex::series() for symbols.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const override

  Implementation of ex::normal() for symbols.
- ex to_rational (exmap &repl) const override

  Implementation of ex::to_rational() for symbols.
- ex to_polynomial (exmap &repl) const override

  Implementation of ex::to_polynomial() for symbols.
- ex conjugate () const override
- ex real_part () const override
- ex imag_part () const override
- bool is_polynomial (const ex &var) const override

  Check whether this is a polynomial in the given variables.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- virtual unsigned get_domain () const
- void set_name (const std::string &n)
- void set_TeX_name (const std::string &n)
- std::string get_name () const
- std::string get_TeX_name () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.

- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for single differentiation of a symbol.
- bool is_equal_same_type (const basic &other) const override

  Returns true if two objects of same type are equal.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- unsigned serial

  unique serial number for comparison
- std::string name

  printname of this symbol
- std::string TeX_name

  LaTeX name of this symbol.

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

Static Private Attributes

- static unsigned next_serial = 0

8.173.1 Detailed Description

Basic CAS symbol.

It has a name because it must know how to output itself.

8.173.2 Constructor & Destructor Documentation

8.173.2.1 symbol() [1/2]

```
GiNaC::symbol::symbol (
            const std::string & initname)  [explicit]
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, name, next_serial, serial, GiNaC::basic::setflag(), and TeX_name.

Referenced by derivative(), GiNaC::matrix::inverse(), is_equal_same_type(), GiNaC::realsymbol::realsymbol(), GiNaC::realsymbol::realsymbol(), GiNaC::realsymbol::realsymbol(), GiNaC::sqrfree_parfrac(), and GiNaC::symbolic_matrix().

8.173.2.2 symbol() [2/2]

```
GiNaC::symbol::symbol (
            const std::string & initname,
            const std::string & texname)
```

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, name, next_serial, serial, GiNaC::basic::setflag(), and TeX_name.

8.173.3 Member Function Documentation

8.173.3.1 info()

```
bool GiNaC::symbol::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References GiNaC::info_flags::cinteger_polynomial, GiNaC::info_flags::crational_polynomial, GiNaC::info_flags::expanded, get_domain(), GiNaC::info_flags::has_indices, GiNaC::info_flags::integer_polynomial, GiNaC::info_flags::nonnegative, GiNaC::info_flags::polynomial, GiNaC::domain::positive, GiNaC::info_flags::positive, GiNaC::info_flags::rational_function, GiNaC::info_flags::rational_polynomial, GiNaC::domain::real, GiNaC::info_flags::real, and GiNaC::info_flags::symbol.

Referenced by GiNaC::conjugate_expl_derivative(), GiNaC::imag_part_expl_derivative(), and GiNaC::real_part_expl_derivative().

8.173.3.2 eval()

```
ex GiNaC::symbol::eval () const  [inline], [override], [virtual]
```

Perform automatic non-interruptive term rewriting rules.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.173.3.3 evalf()

```
ex GiNaC::symbol::evalf () const  [inline], [override], [virtual]
```

Evaluate object numerically.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.173.3.4 series()

```
ex GiNaC::symbol::series (
            const relational & r,
            int order,
            unsigned options = 0) const  [override], [virtual]
```

Implementation of ex::series() for symbols.

See also

ex::series

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::expair::expair(), GINAC_ASSERT, GiNaC::is_a(), is_equal_same_type(), GiNaC::ex::is_zero(), GiNaC::relational::lhs(), GiNaC::numeric::numeric(), GiNaC::pseries::pseries(), and GiNaC::relational::rhs().

8.173.3.5 subs()

```
ex GiNaC::symbol::subs (
            const exmap & m,
            unsigned options = 0) const  [inline], [override], [virtual]
```

Substitute a set of objects by arbitrary expressions.

The ex returned will already be evaluated.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, and GiNaC::basic::subs_one_level().

8.173.3.6 normal()

```
ex GiNaC::symbol::normal (
            exmap & repl,
            exmap & rev_lookup,
            lst & modifier) const  [override], [virtual]
```

Implementation of ex::normal() for symbols.

This returns the unmodified symbol.

See also

ex::normal

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::dynallocate(), and GiNaC::basic::ex.

8.173.3.7 to_rational()

```
ex GiNaC::symbol::to_rational (
            exmap & repl) const  [override], [virtual]
```

Implementation of ex::to_rational() for symbols.

This returns the unmodified symbol.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.173.3.8 to_polynomial()

```
ex GiNaC::symbol::to_polynomial (
            exmap & repl) const  [override], [virtual]
```

Implementation of ex::to_polynomial() for symbols.

This returns the unmodified symbol.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.173.3.9 conjugate()

```
ex GiNaC::symbol::conjugate () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.173.3.10 real_part()

```
ex GiNaC::symbol::real_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.173.3.11 imag_part()

```
ex GiNaC::symbol::imag_part () const  [override], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.173.3.12 is_polynomial()

```
bool GiNaC::symbol::is_polynomial (
            const ex & var) const  [override], [virtual]
```

Check whether this is a polynomial in the given variables.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex.

8.173.3.13 archive()

```
void GiNaC::symbol::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

Archive the object.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_string(), name, and TeX_name.

8.173.3.14 read_archive()

```
void GiNaC::symbol::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

Read object from archive_node.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References GiNaC::container< C >::append(), GiNaC::status_flags::dynallocated, GiNaC::status_flags::evaluated, GiNaC::ex_to(), GiNaC::status_flags::expanded, GiNaC::archive_node::find_string(), GiNaC::is_a(), name, next_serial, serial, GiNaC::basic::setflag(), and TeX_name.

8.173.3.15 derivative()

```
ex GiNaC::symbol::derivative (
            const symbol & s) const  [override], [protected], [virtual]
```

Implementation of ex::diff() for single differentiation of a symbol.

It returns 1 or 0.

See also

ex::diff

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1, GiNaC::basic::compare_same_type(), GiNaC::basic::ex, and symbol().

8.173.3.16 is_equal_same_type()

```
bool GiNaC::symbol::is_equal_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if two objects of same type are equal.

Normally needs not be reimplemented as long as it wasn't overwritten by some parent class, since it just calls compare_same_type(). The reason why this function exists is that sometimes it is easier to determine equality than an order relation and then it can be overridden.

Reimplemented from GiNaC::basic.

References GiNaC::basic::basic(), GINAC_ASSERT, GiNaC::is_a(), serial, and symbol().

Referenced by series().

8.173.3.17 calchash()

```
unsigned GiNaC::symbol::calchash () const  [override], [protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References GiNaC::golden_ratio_hash(), GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, serial, and GiNaC::basic::setflag().

8.173.3.18 get_domain()

```
virtual unsigned GiNaC::symbol::get_domain () const  [inline], [virtual]
```

Reimplemented in GiNaC::possymbol, and GiNaC::realsymbol.

References GiNaC::domain::complex.

Referenced by do_print_tree(), and info().

8.173.3.19 set_name()

```
void GiNaC::symbol::set_name (
            const std::string & n)  [inline]
```

References name.

8.173.3.20 set_TeX_name()

```
void GiNaC::symbol::set_TeX_name (
            const std::string & n)  [inline]
```

References TeX_name.

8.173.3.21 get_name()

```
std::string GiNaC::symbol::get_name () const
```

References name, and serial.

Referenced by do_print(), and get_TeX_name().

8.173.3.22 get_TeX_name()

```
std::string GiNaC::symbol::get_TeX_name () const
```

References GiNaC::get_default_TeX_name(), get_name(), and TeX_name.

8.173.3.23 do_print()

```
void GiNaC::symbol::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References get_name(), and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.173.3.24 do_print_latex()

```
void GiNaC::symbol::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

References GiNaC::get_default_TeX_name(), name, GiNaC::print_context::s, serial, and TeX_name.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.173.3.25 do_print_tree()

```
void GiNaC::symbol::do_print_tree (
             const print_tree & c,
             unsigned level) const  [protected]
```

References GiNaC::basic::flags, get_domain(), GiNaC::basic::hashvalue, name, GiNaC::print_context::s, and serial.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.173.3.26 do_print_python_repr()

```
void GiNaC::symbol::do_print_python_repr (
             const print_python_repr & c,
             unsigned level) const  [protected]
```

References name, GiNaC::print_context::s, serial, and TeX_name.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.173.4 Member Data Documentation

8.173.4.1 serial

```
unsigned GiNaC::symbol::serial  [protected]
```

unique serial number for comparison

Referenced by calchash(), do_print_latex(), do_print_python_repr(), do_print_tree(), get_name(), is_equal_same_type(), read_archive(), symbol(), and symbol().

8.173.4.2 name

```
std::string GiNaC::symbol::name  [mutable], [protected]
```

printname of this symbol

Referenced by archive(), do_print_latex(), do_print_python_repr(), do_print_tree(), get_name(), read_archive(), set_name(), symbol(), and symbol().

8.173.4.3 TeX_name

```
std::string GiNaC::symbol::TeX_name  [protected]
```

LaTeX name of this symbol.

Referenced by archive(), do_print_latex(), do_print_python_repr(), get_TeX_name(), read_archive(), set_TeX_name(), symbol(), and symbol().

8.173.4.4 next_serial

```
unsigned GiNaC::symbol::next_serial = 0  [static], [private]
```

Referenced by GiNaC::GINAC_BIND_UNARCHIVER(), read_archive(), symbol(), and symbol().

The documentation for this class was generated from the following files:

- symbol.h
- normal.cpp
- pseries.cpp
- symbol.cpp

8.174 GiNaC::symbolset Class Reference

Public Member Functions

- symbolset (const ex &e)
- bool has (const ex &e) const

Private Member Functions

- void insert_symbols (const ex &e)

Private Attributes

- exset s

8.174.1 Constructor & Destructor Documentation

8.174.1.1 symbolset()

```
GiNaC::symbolset::symbolset (
            const ex & e)  [inline], [explicit]
```

References insert_symbols().

8.174.2 Member Function Documentation

8.174.2.1 insert_symbols()

```
void GiNaC::symbolset::insert_symbols (
            const ex & e)  [inline], [private]
```

References insert_symbols(), GiNaC::is_a(), and s.

Referenced by insert_symbols(), and symbolset().

8.174.2.2 has()

```
bool GiNaC::symbolset::has (
            const ex & e) const  [inline]
```

References s.

Referenced by GiNaC::lsolve().

8.174.3 Member Data Documentation

8.174.3.1 s

```
exset GiNaC::symbolset::s  [private]
```

Referenced by has(), and insert_symbols().

The documentation for this class was generated from the following file:

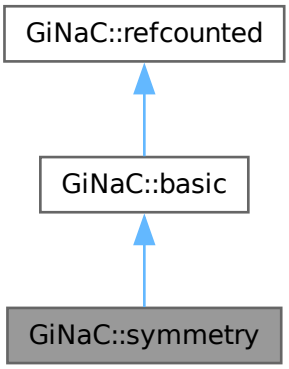
- inifcns.cpp

8.175 GiNaC::symmetry Class Reference

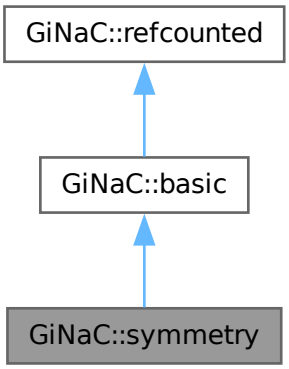
This class describes the symmetry of a group of indices.

`#include <symmetry.h>`

Inheritance diagram for GiNaC::symmetry:



Collaboration diagram for GiNaC::symmetry:



Public Types

- enum symmetry_type { none , symmetric , antisymmetric , cyclic }

  Type of symmetry.

Public Member Functions

- symmetry (unsigned i)

  Create leaf node that represents one index.
- symmetry (symmetry_type t, const symmetry &c1, const symmetry &c2)

  Create node with two children.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- symmetry_type get_type () const

  Get symmetry type.
- void set_type (symmetry_type t)

  Set symmetry type.
- symmetry & add (const symmetry &c)

  Add child node, check index sets for consistency.
- void validate (unsigned n)

  Verify that all indices of this node are in the range [0..n-1].
- bool has_symmetry () const

  Check whether this node actually represents any kind of symmetry.
- bool has_nonsymmetric () const

  Check whether this node involves anything non symmetric.
- bool has_cyclic () const

  Check whether this node involves a cyclic symmetry.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.

- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.

- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Attributes

- symmetry_type type

  Type of symmetry described by this node.
- std::set< unsigned > indices

  Sorted union set of all indices handled by this node.
- exvector children

  Vector of child nodes.

Friends

- class sy_is_less
- class sy_swap
- int canonicalize (exvector::iterator v, const symmetry &symm)

  Canonicalize the order of elements of an expression vector, according to the symmetry properties defined in a symmetry tree.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.175.1 Detailed Description

This class describes the symmetry of a group of indices.

These objects can be grouped into a tree to form complex mixed symmetries.

8.175.2 Member Enumeration Documentation

8.175.2.1 symmetry_type

```
enum GiNaC::symmetry::symmetry_type
```

Type of symmetry.

Enumerator

none	no symmetry properties
symmetric	totally symmetric
antisymmetric	totally antisymmetric
cyclic	cyclic symmetry

8.175.3 Constructor & Destructor Documentation

8.175.3.1 symmetry() [1/2]

```
GiNaC::symmetry::symmetry (
            unsigned i)
```

Create leaf node that represents one index.

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, indices, none, GiNaC::basic::setflag(), symmetry(), and type.

Referenced by add(), canonicalize, GiNaC::sy_anti(), GiNaC::sy_anti(), GiNaC::sy_anti(), GiNaC::sy_cycl(), GiNaC::sy_cycl(), GiNaC::sy_cycl(), GiNaC::sy_none(), GiNaC::sy_none(), GiNaC::sy_none(), GiNaC::sy_none(), sy_swap, GiNaC::sy_symm(), GiNaC::sy_symm(), GiNaC::sy_symm(), symmetry(), and symmetry().

8.175.3.2 symmetry() [2/2]

```
GiNaC::symmetry::symmetry (
            symmetry_type t,
            const symmetry & c1,
            const symmetry & c2)
```

Create node with two children.

References GiNaC::add::add(), GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::basic::setflag(), symmetry(), and type.

8.175.4 Member Function Documentation

8.175.4.1 archive()

```
void GiNaC::symmetry::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

Archive the object.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_ex(), GiNaC::archive_node::add_unsigned(), children, indices, and type.

8.175.4.2 read_archive()

```
void GiNaC::symmetry::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

Construct object from archive_node.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References GiNaC::add::add(), GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::archive_node::find_ex(), GiNaC::archive_node::find_unsigr
indices, and type.

8.175.4.3 calchash()

```
unsigned GiNaC::symmetry::calchash () const  [override], [protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References children, GiNaC::status_flags::evaluated, GiNaC::basic::flags, GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, indices, none, GiNaC::rotate_left(), GiNaC::basic::setflag(), and type.

8.175.4.4 get_type()

```
symmetry_type GiNaC::symmetry::get_type () const  [inline]
```

Get symmetry type.

References type.

8.175.4.5 set_type()

```
void GiNaC::symmetry::set_type (
            symmetry_type t)  [inline]
```

Set symmetry type.

References type.

Referenced by GiNaC::sy_anti(), GiNaC::sy_cycl(), and GiNaC::sy_symm().

8.175.4.6 add()

```
symmetry & GiNaC::symmetry::add (
            const symmetry & c)
```

Add child node, check index sets for consistency.

References children, GiNaC::ex_to(), GINAC_ASSERT, indices, GiNaC::is_exactly_a(), none, symmetry(), and type.

Referenced by GiNaC::sy_anti(), GiNaC::sy_anti(), GiNaC::sy_cycl(), GiNaC::sy_cycl(), GiNaC::sy_none(), GiNaC::sy_none(), GiNaC::sy_symm(), and GiNaC::sy_symm().

8.175.4.7 validate()

```
void GiNaC::symmetry::validate (
            unsigned n)
```

Verify that all indices of this node are in the range [0..n-1].

This function throws an exception if the verification fails. If the top node has a type != none and no children, add all indices in the range [0..n-1] as children.

References GiNaC::add::add(), indices, none, and type.

8.175.4.8 has_symmetry()

```
bool GiNaC::symmetry::has_symmetry () const  [inline]
```

Check whether this node actually represents any kind of symmetry.

References children, none, and type.

8.175.4.9 has_nonsymmetric()

```
bool GiNaC::symmetry::has_nonsymmetric () const
```

Check whether this node involves anything non symmetric.

References antisymmetric, children, cyclic, GiNaC::ex_to(), has_nonsymmetric(), and type.

Referenced by has_nonsymmetric().

8.175.4.10 has_cyclic()

```
bool GiNaC::symmetry::has_cyclic () const
```

Check whether this node involves a cyclic symmetry.

References children, cyclic, GiNaC::ex_to(), has_cyclic(), and type.

Referenced by has_cyclic().

8.175.4.11 do_print()

```
void GiNaC::symmetry::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References antisymmetric, children, cyclic, indices, none, GiNaC::print_context::s, symmetric, and type.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.175.4.12 do_print_tree()

```
void GiNaC::symmetry::do_print_tree (
             const print_tree & c,
             unsigned level) const  [protected]
```

References antisymmetric, children, cyclic, GiNaC::print_tree::delta_indent, GiNaC::basic::flags, GiNaC::basic::hashvalue, indices, none, GiNaC::print_context::s, symmetric, and type.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.175.5 Friends And Related Symbol Documentation

8.175.5.1 sy_is_less

```
friend class sy_is_less  [friend]
```

References sy_is_less.

Referenced by sy_is_less.

8.175.5.2 sy_swap

```
friend class sy_swap  [friend]
```

References GiNaC::basic::basic(), canonicalize, GINAC_DECLARE_REGISTERED_CLASS, sy_swap, GiNaC::symm(), and symmetry().

Referenced by sy_swap.

8.175.5.3 canonicalize

```
int canonicalize (
            exvector::iterator v,
            const symmetry & symm)  [friend]
```

Canonicalize the order of elements of an expression vector, according to the symmetry properties defined in a symmetry tree.

Parameters

v	Start of expression vector
symm	Root node of symmetry tree

Returns

the overall sign introduced by the reordering (+1, -1 or 0) or numeric_limits<int>::max() if nothing changed

References antisymmetric, GiNaC::ex::begin(), canonicalize, GiNaC::class_info< OPT >::first, cyclic, GiNaC::cyclic_permutation(), GiNaC::ex::end(), GiNaC::ex_to(), GINAC_ASSERT, GiNaC::is_exactly_a(), GiNaC::permutation_sign(), GiNaC::shaker_sort(), GiNaC::sy_is_less::sy_is_less(), GiNaC::sy_swap::sy_swap(), GiNaC::symm(), symmetric, and symmetry().

Referenced by canonicalize, and sy_swap.

8.175.6 Member Data Documentation

8.175.6.1 type

```
symmetry_type GiNaC::symmetry::type  [private]
```

Type of symmetry described by this node.

Referenced by add(), archive(), calchash(), do_print(), do_print_tree(), get_type(), has_cyclic(), has_nonsymmetric(), has_symmetry(), read_archive(), set_type(), symmetry(), symmetry(), and validate().

8.175.6.2 indices

```
std::set<unsigned> GiNaC::symmetry::indices  [private]
```

Sorted union set of all indices handled by this node.

Referenced by add(), archive(), calchash(), do_print(), do_print_tree(), read_archive(), symmetry(), and validate().

8.175.6.3 children

```
exvector GiNaC::symmetry::children  [private]
```

Vector of child nodes.

Referenced by add(), archive(), calchash(), do_print(), do_print_tree(), has_cyclic(), has_nonsymmetric(), and has_symmetry().

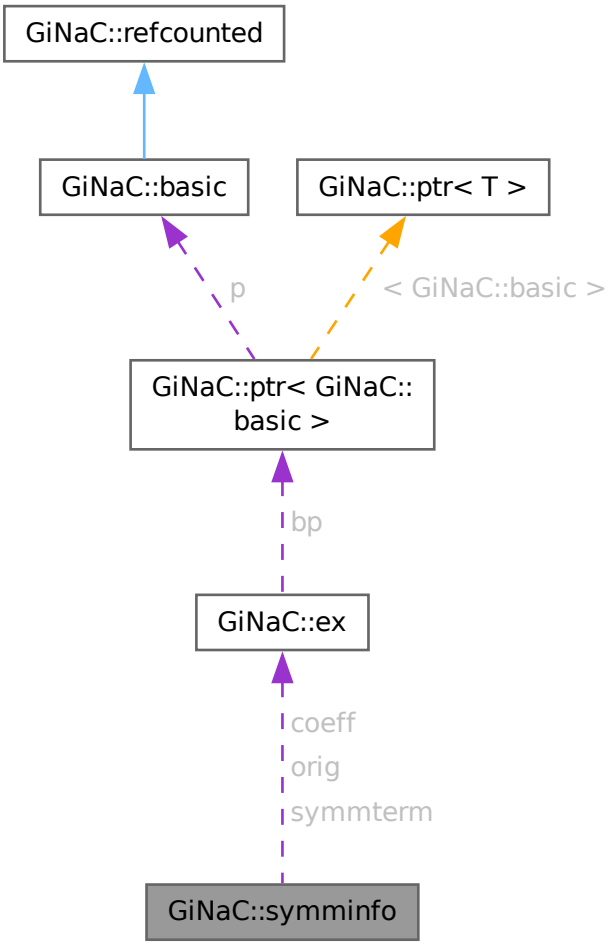
The documentation for this class was generated from the following files:

- symmetry.h
- symmetry.cpp

8.176 GiNaC::symminfo Class Reference

This structure stores the individual symmetrized terms obtained during the simplification of sums.

Collaboration diagram for GiNaC::symminfo:



Public Member Functions

- symminfo ()
- symminfo (const ex &symmterm_, const ex &orig_, size_t num_)

Public Attributes

- ex symmterm
  symmetrized term
- ex coeff
  coefficient of symmetrized term
- ex orig
  original term
- size_t num
  how many symmetrized terms resulted from the original term

8.176.1 Detailed Description

This structure stores the individual symmetrized terms obtained during the simplification of sums.

8.176.2 Constructor & Destructor Documentation

8.176.2.1 symminfo() [1/2]

```
GiNaC::symminfo::symminfo ()  [inline]
```

References num.

Referenced by GiNaC::indexed::simplify_indexed.

8.176.2.2 symminfo() [2/2]

```
GiNaC::symminfo::symminfo (
            const ex & symmterm_,
            const ex & orig_,
            size_t num_)  [inline]
```

References GiNaC::is_exactly_a(), GiNaC::ex::nops(), num, GiNaC::ex::op(), and orig.

8.176.3 Member Data Documentation

8.176.3.1 symmterm

```
ex GiNaC::symminfo::symmterm
```

symmetrized term

Referenced by GiNaC::symminfo_is_less_by_symmterm::operator()().

8.176.3.2 coeff

```
ex GiNaC::symminfo::coeff
```

coefficient of symmetrized term

8.176.3.3 orig

```
ex GiNaC::symminfo::orig
```

original term

Referenced by GiNaC::symminfo_is_less_by_orig::operator()(), and symminfo().

8.176.3.4 num

```
size_t GiNaC::symminfo::num
```

how many symmetrized terms resulted from the original term

Referenced by symminfo(), and symminfo().

The documentation for this class was generated from the following file:

- indexed.cpp

8.177 GiNaC::symminfo_is_less_by_orig Class Reference

Public Member Functions

- bool operator() (const symminfo &si1, const symminfo &si2) const

8.177.1 Member Function Documentation

8.177.1.1 operator()()

```
bool GiNaC::symminfo_is_less_by_orig::operator() (
            const symminfo & si1,
            const symminfo & si2) const  [inline]
```

References GiNaC::ex::compare(), and GiNaC::symminfo::orig.

The documentation for this class was generated from the following file:

- indexed.cpp

8.178 GiNaC::symminfo_is_less_by_symmterm Class Reference

Public Member Functions

- bool operator() (const symminfo &si1, const symminfo &si2) const

8.178.1 Member Function Documentation

8.178.1.1 operator()()

```
bool GiNaC::symminfo_is_less_by_symmterm::operator() (
            const symminfo & si1,
            const symminfo & si2) const  [inline]
```

References GiNaC::ex::compare(), and GiNaC::symminfo::symmterm.

The documentation for this class was generated from the following file:

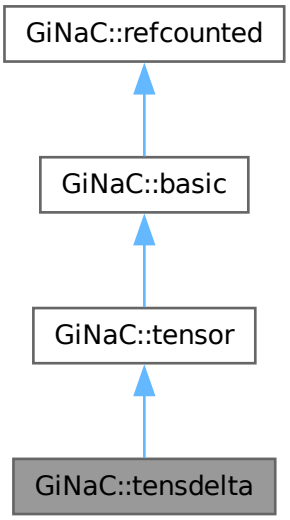
- indexed.cpp

8.179 GiNaC::tensdelta Class Reference

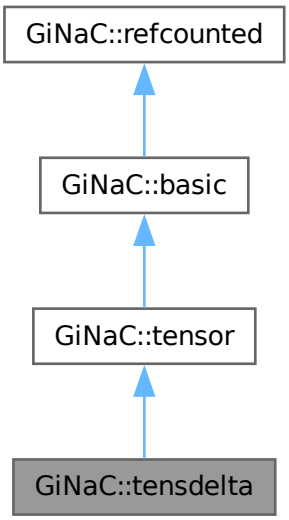
This class represents the delta tensor.

```
#include <tensor.h>
```

Inheritance diagram for GiNaC::tensdelta:



Collaboration diagram for GiNaC::tensdelta:



Public Member Functions

- bool info (unsigned inf) const override

  Information about the object.
- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of an indexed delta tensor.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of an indexed delta tensor with something else.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.

- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().

- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- unsigned return_type () const override
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.179.1 Detailed Description

This class represents the delta tensor.

If indexed, it must have exactly two indices of the same type.

8.179.2 Member Function Documentation

8.179.2.1 info()

```
bool GiNaC::tensdelta::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References info(), and GiNaC::info_flags::real.

Referenced by info().

8.179.2.2 eval_indexed()

```
ex GiNaC::tensdelta::eval_indexed (
            const basic & i) const  [override], [virtual]
```

Automatic symbolic evaluation of an indexed delta tensor.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::_ex1, eval_indexed(), GiNaC::ex_to(), GiNaC::idx::get_dim(), GiNaC::idx::get_value(), GINAC_ASSERT, GiNaC::info_flags::integer, GiNaC::is_a(), GiNaC::is_dummy_pair(), GiNaC::ex::is_equal(), GiNaC::idx::minimal_dim(), GiNaC::subs_options::no_pattern, GiNaC::idx::replace_dim(), and GiNaC::ex::subs().

Referenced by eval_indexed().

8.179.2.3 contract_with()

```
bool GiNaC::tensdelta::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [override], [virtual]
```

Contraction of an indexed delta tensor with something else.

Reimplemented from GiNaC::basic.

References GINAC_ASSERT, GiNaC::is_a(), and GiNaC::tensor::replace_contr_index().

8.179.2.4 return_type()

```
unsigned GiNaC::tensdelta::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative.

8.179.2.5 do_print()

```
void GiNaC::tensdelta::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

8.179.2.6 do_print_latex()

```
void GiNaC::tensdelta::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

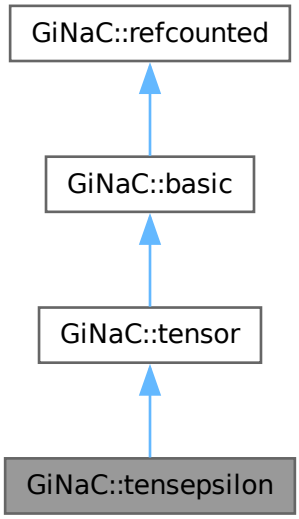
- tensor.h
- tensor.cpp

8.180 GiNaC::tensepsilon Class Reference

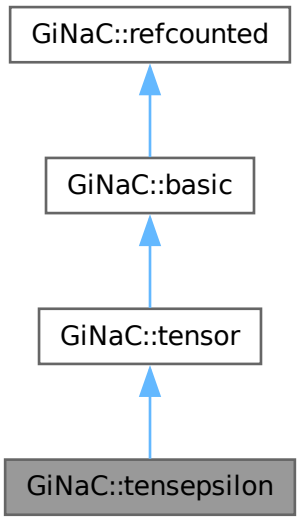
This class represents the totally antisymmetric epsilon tensor.

```
#include <tensor.h>
```

Inheritance diagram for GiNaC::tensepsilon:



Collaboration diagram for GiNaC::tensepsilon:



Public Member Functions

- tensepsilon (bool minkowski, bool pos_sig)
- bool info (unsigned inf) const override

  Information about the object.
- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of an indexed epsilon tensor.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of epsilon tensor with something else.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.
- basic (const basic &other)
- const basic & operator= (const basic &other)

basic assignment operator: the other object might be of a derived class.

- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

Default implementation of ex::series().

- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- unsigned return_type () const override
- void do_print (const print_context &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Attributes

- bool minkowski

  If true, tensor is in Minkowski-type space.
- bool pos_sig

  If true, the metric is assumed to be diag(-1,1,1...).

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.180.1 Detailed Description

This class represents the totally antisymmetric epsilon tensor.

If indexed, all indices must be of the same type and their number must be equal to the dimension of the index space.

8.180.2 Constructor & Destructor Documentation

8.180.2.1 tensepsilon()

```
GiNaC::tensepsilon::tensepsilon (
            bool minkowski,
            bool pos_sig)
```

References minkowski, and pos_sig.

8.180.3 Member Function Documentation

8.180.3.1 info()

```
bool GiNaC::tensepsilon::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References GiNaC::info_flags::real.

8.180.3.2 eval_indexed()

```
ex GiNaC::tensepsilon::eval_indexed (
            const basic & i) const  [override], [virtual]
```

Automatic symbolic evaluation of an indexed epsilon tensor.

Reimplemented from GiNaC::basic.

References GiNaC::_ex0, GiNaC::ex_to(), GINAC_ASSERT, GiNaC::basic::hold(), GiNaC::is_a(), GiNaC::is_zero(), minkowski, GiNaC::info_flags::nonnegint, GiNaC::basic::nops(), GiNaC::basic::op(), GiNaC::permutation_sign(), and pos_sig.

8.180.3.3 contract_with()

```
bool GiNaC::tensepsilon::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [override], [virtual]
```

Contraction of epsilon tensor with something else.

Reimplemented from GiNaC::basic.

References GiNaC::_ex1, GiNaC::delta_tensor(), GiNaC::matrix::determinant(), GINAC_ASSERT, GiNaC::is_a(), GiNaC::is_exactly_a(), GiNaC::lorentz_g(), GiNaC::metric_tensor(), minkowski, pos_sig, and GiNaC::ex::simplify_indexed().

8.180.3.4 archive()

```
void GiNaC::tensepsilon::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_bool(), minkowski, and pos_sig.

8.180.3.5 read_archive()

```
void GiNaC::tensepsilon::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::find_bool(), minkowski, and pos_sig.

8.180.3.6 return_type()

```
unsigned GiNaC::tensepsilon::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative.

8.180.3.7 do_print()

```
void GiNaC::tensepsilon::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

8.180.3.8 do_print_latex()

```
void GiNaC::tensepsilon::do_print_latex (
             const print_latex & c,
             unsigned level) const  [protected]
```

8.180.4 Member Data Documentation

8.180.4.1 minkowski

```
bool GiNaC::tensepsilon::minkowski  [private]
```

If true, tensor is in Minkowski-type space.

Otherwise it is in a Euclidean space.

Referenced by archive(), contract_with(), eval_indexed(), read_archive(), and tensepsilon().

8.180.4.2 pos_sig

```
bool GiNaC::tensepsilon::pos_sig  [private]
```

If true, the metric is assumed to be diag(-1,1,1...).

Otherwise it is diag(1,-1,-1,...). This is only relevant if minkowski = true.

Referenced by archive(), contract_with(), eval_indexed(), read_archive(), and tensepsilon().

The documentation for this class was generated from the following files:

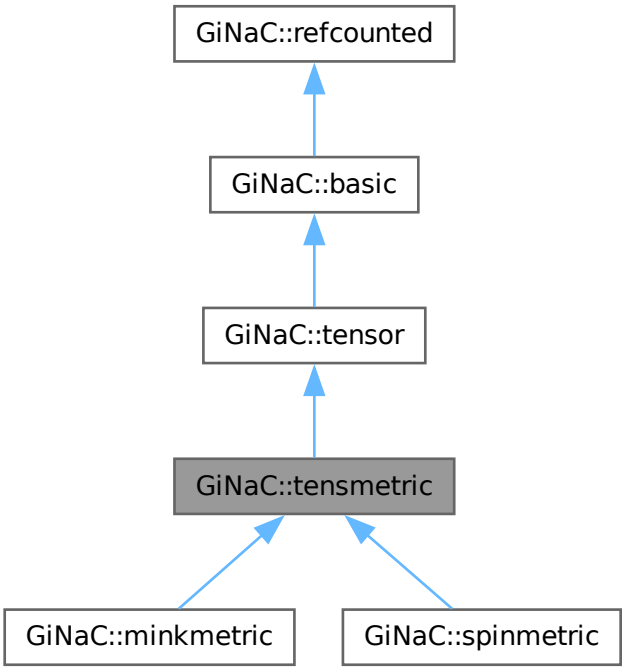
- tensor.h
- tensor.cpp

8.181 GiNaC::tensmetric Class Reference

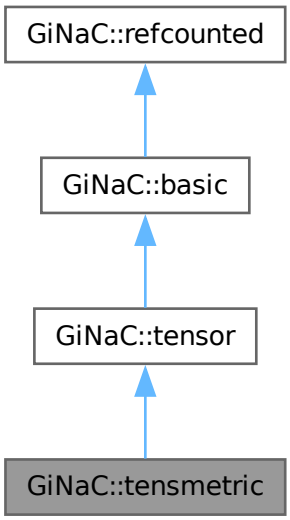
This class represents a general metric tensor which can be used to raise/lower indices.

`#include <tensor.h>`

Inheritance diagram for GiNaC::tensmetric:



Collaboration diagram for GiNaC::tensmetric:



Public Member Functions

- bool info (unsigned inf) const override

  Information about the object.
- ex eval_indexed (const basic &i) const override

  Automatic symbolic evaluation of an indexed metric tensor.
- bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const override

  Contraction of an indexed metric tensor with something else.

Public Member Functions inherited from GiNaC::tensor

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.

- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().

- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- unsigned return_type () const override
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::tensor

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.181.1 Detailed Description

This class represents a general metric tensor which can be used to raise/lower indices.

If indexed, it must have exactly two indices of the same type which must be of class varidx or a subclass.

8.181.2 Member Function Documentation

8.181.2.1 info()

```
bool GiNaC::tensmetric::info (
            unsigned inf) const  [override], [virtual]
```

Information about the object.

See also

class info_flags

Reimplemented from GiNaC::basic.

References GiNaC::info_flags::real.

8.181.2.2 eval_indexed()

```
ex GiNaC::tensmetric::eval_indexed (
            const basic & i) const  [override], [virtual]
```

Automatic symbolic evaluation of an indexed metric tensor.

Reimplemented from GiNaC::basic.

References GiNaC::delta_tensor(), GiNaC::ex_to(), GiNaC::idx::get_dim(), GINAC_ASSERT, GiNaC::basic::hold(), GiNaC::is_a(), GiNaC::varidx::is_covariant(), GiNaC::ex::is_equal(), GiNaC::idx::minimal_dim(), GiNaC::subs_options::no_pattern, GiNaC::basic::nops(), GiNaC::basic::op(), GiNaC::idx::replace_dim(), and GiNaC::basic::subs().

8.181.2.3 contract_with()

```
bool GiNaC::tensmetric::contract_with (
            exvector::iterator self,
            exvector::iterator other,
            exvector & v) const  [override], [virtual]
```

Contraction of an indexed metric tensor with something else.

Reimplemented from GiNaC::basic.

References GINAC_ASSERT, GiNaC::is_a(), and GiNaC::tensor::replace_contr_index().

8.181.2.4 return_type()

```
unsigned GiNaC::tensmetric::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::commutative.

8.181.2.5 do_print()

```
void GiNaC::tensmetric::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

The documentation for this class was generated from the following files:

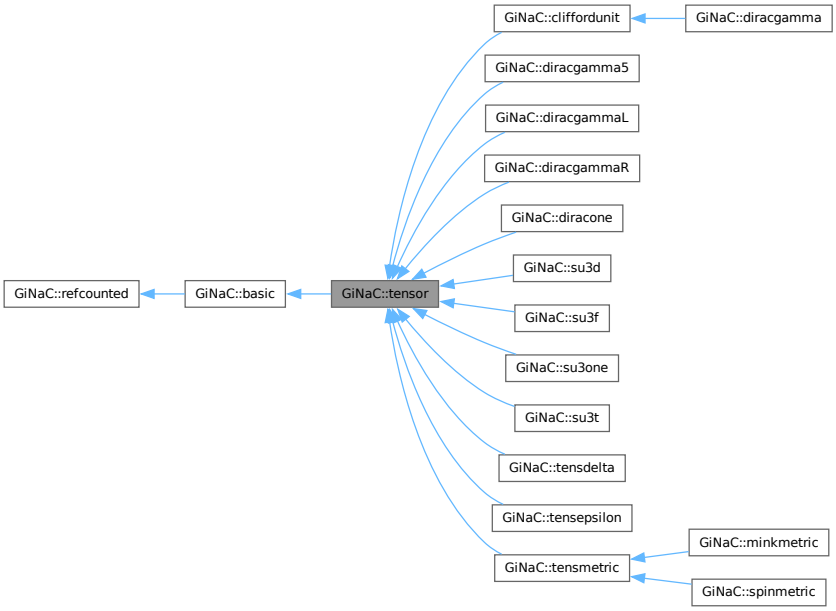
- tensor.h
- tensor.cpp

8.182 GiNaC::tensor Class Reference

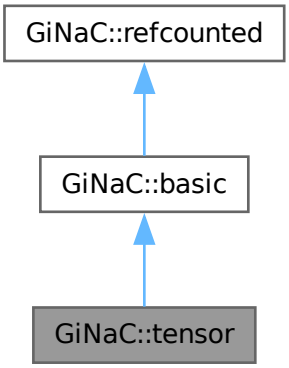
This class holds one of GiNaC's predefined special tensors such as the delta and the metric tensors.

```
#include <tensor.h>
```

Inheritance diagram for GiNaC::tensor:



Collaboration diagram for GiNaC::tensor:



Public Member Functions

- bool replace_contr_index (exvector::iterator self, exvector::iterator other) const

  Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

Little wrapper around printtree to be called within a debugger.

- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

Add two indexed expressions.

- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

  Load (deserialize) the object from an archive node.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- unsigned return_type () const override

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.182.1 Detailed Description

This class holds one of GiNaC's predefined special tensors such as the delta and the metric tensors.

They are represented without indices. To attach indices to them, wrap them in an object of class indexed.

8.182.2 Member Function Documentation

8.182.2.1 return_type()

```
unsigned GiNaC::tensor::return_type () const  [inline], [override], [protected], [virtual]
```

Reimplemented from GiNaC::basic.

References GiNaC::return_types::noncommutative_composite.

8.182.2.2 replace_contr_index()

```
bool GiNaC::tensor::replace_contr_index (
             exvector::iterator self,
             exvector::iterator other) const
```

Replace dummy index in contracted-with object by the contracting object's second index (used internally for delta and metric tensor contractions.

References GiNaC::_ex1, GiNaC::basic::ex, GiNaC::ex_to(), GiNaC::is_a(), GiNaC::is_dummy_pair(), GiNaC::idx::is_symbolic(), GiNaC::idx::minimal_dim(), GiNaC::idx::replace_dim(), and GiNaC::ex::subs().

Referenced by GiNaC::tensdelta::contract_with(), and GiNaC::tensmetric::contract_with().

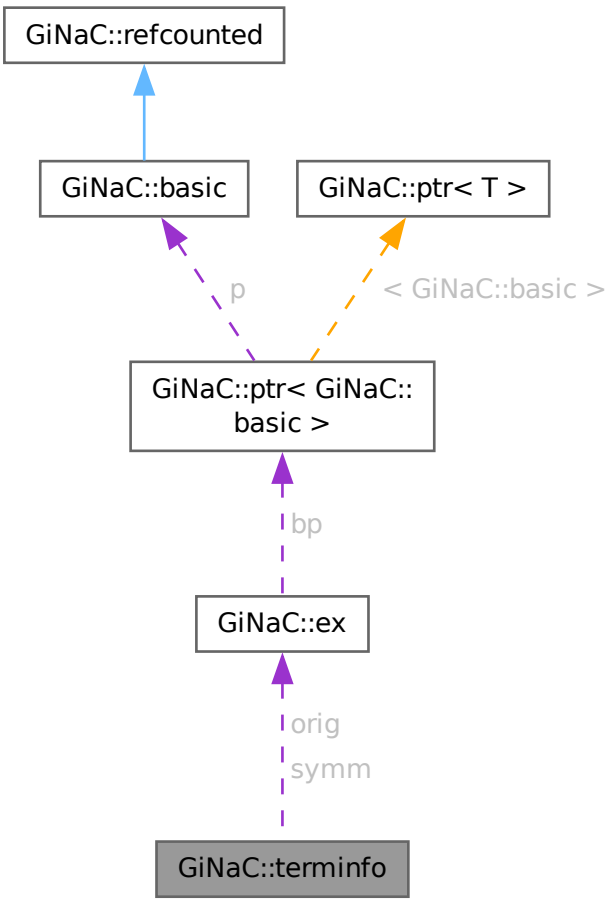
The documentation for this class was generated from the following files:

- tensor.h
- tensor.cpp

8.183 GiNaC::terminfo Class Reference

This structure stores the original and symmetrized versions of terms obtained during the simplification of sums.

Collaboration diagram for GiNaC::terminfo:



Public Member Functions

- terminfo (const ex &orig_, const ex &symm_)

Public Attributes

- ex orig
  original term
- ex symm
  symmetrized term

8.183.1 Detailed Description

This structure stores the original and symmetrized versions of terms obtained during the simplification of sums.

8.183.2 Constructor & Destructor Documentation

8.183.2.1 terminfo()

```
GiNaC::terminfo::terminfo (
            const ex & orig_,
            const ex & symm_)  [inline]
```

References orig, and symm.

Referenced by GiNaC::indexed::simplify_indexed.

8.183.3 Member Data Documentation

8.183.3.1 orig

```
ex GiNaC::terminfo::orig
```

original term

Referenced by terminfo().

8.183.3.2 symm

```
ex GiNaC::terminfo::symm
```

symmetrized term

Referenced by GiNaC::terminfo_is_less::operator()(), and terminfo().

The documentation for this class was generated from the following file:

- indexed.cpp

8.184 GiNaC::terminfo_is_less Class Reference

Public Member Functions

- bool operator() (const terminfo &ti1, const terminfo &ti2) const

8.184.1 Member Function Documentation

8.184.1.1 operator()()

```
bool GiNaC::terminfo_is_less::operator() (
            const terminfo & ti1,
            const terminfo & ti2) const  [inline]
```

References GiNaC::ex::compare(), and GiNaC::terminfo::symm.

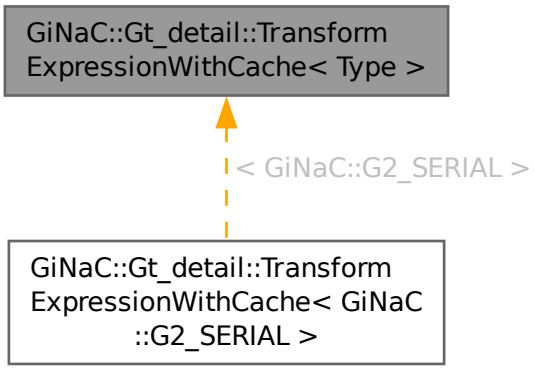
The documentation for this class was generated from the following file:

- indexed.cpp

8.185 GiNaC::Gt_detail::TransformExpressionWithCache< Type > Class Template Reference

```
#include <Gt_helpers.h>
```

Inheritance diagram for GiNaC::Gt_detail::TransformExpressionWithCache< Type >:



Public Member Functions

- TransformExpressionWithCache (const std::function< ex(const ex &)> &func_obj, const std::function< ex(const ex &)> &func_default=[](const ex &obj) -> ex { return obj;})
- ex operator() (const ex &input)
- void print_cache () const
- const exhashmap< ex > & get_cache () const
- void clear_cache ()

Private Member Functions

- ex impl (const ex &expr)

Static Private Member Functions

- template<typename T = Type>
  static std::enable_if< std::is_base_of< basic, T >::value, bool >::type check_type (const ex &expr)
- template<typename T = Type>
  static std::enable_if<!std::is_base_of< basic, T >::value, bool >::type check_type (const ex &expr)

Private Attributes

- const std::function< ex(const ex &)> func_obj
- const std::function< ex(const ex &)> func_default
- exhashmap< ex > cache

8.185.1 Constructor & Destructor Documentation

8.185.1.1 TransformExpressionWithCache()

```
template<typename Type>
GiNaC::Gt_detail::TransformExpressionWithCache< Type >::TransformExpressionWithCache (
            const std::function< ex(const ex &)> & func_obj,
            const std::function< ex(const ex &)> & func_default = [](const ex& obj) -> ex { return obj; })
[inline]
```

References func_default, and func_obj.

8.185.2 Member Function Documentation

8.185.2.1 operator()()

```
template<typename Type>
ex GiNaC::Gt_detail::TransformExpressionWithCache< Type >::operator() (
            const ex & input)  [inline]
```

References impl().

8.185.2.2 print_cache()

```
template<typename Type>
void GiNaC::Gt_detail::TransformExpressionWithCache< Type >::print_cache () const
```

References cache.

8.185.2.3 get_cache()

```
template<typename Type>
const exhashmap< ex > & GiNaC::Gt_detail::TransformExpressionWithCache< Type >::get_cache ()
const  [inline]
```

References cache.

8.185.2.4 clear_cache()

```
template<typename Type>
void GiNaC::Gt_detail::TransformExpressionWithCache< Type >::clear_cache ()  [inline]
```

References cache.

8.185.2.5 check_type() [1/2]

```
template<typename Type>
template<typename T>
std::enable_if<!std::is_base_of< basic, T >::value, bool >::type GiNaC::Gt_detail::TransformExpressionWithCa
Type >::check_type (
            const ex & expr)  [static], [private]
```

References GiNaC::is_exactly_a().

Referenced by impl().

8.185.2.6 check_type() [2/2]

```
template<typename Type>
template<typename T = Type>
std::enable_if<!std::is_base_of< basic, T >::value, bool >::type GiNaC::Gt_detail::TransformExpressionWithCa
Type >::check_type (
            const ex & expr)  [static], [private]
```

References impl().

8.185.2.7 impl()

```
template<typename Type>
ex GiNaC::Gt_detail::TransformExpressionWithCache< Type >::impl (
            const ex & expr)  [private]
```

References GiNaC::add::add(), GiNaC::container< C >::append(), cache, check_type(), func_default, func_obj, impl(), GiNaC::is_exactly_a(), GiNaC::mul::mul(), GiNaC::ex::nops(), GiNaC::ex::op(), and GiNaC::power::power().

Referenced by check_type(), impl(), and operator()().

8.185.3 Member Data Documentation

8.185.3.1 func_obj

```
template<typename Type>
const std::function<ex(const ex&)> GiNaC::Gt_detail::TransformExpressionWithCache< Type >::↩
func_obj  [private]
```

Referenced by impl(), and TransformExpressionWithCache().

8.185.3.2 func_default

```
template<typename Type>
const std::function<ex(const ex&)> GiNaC::Gt_detail::TransformExpressionWithCache< Type >::↩
func_default  [private]
```

Referenced by impl(), and TransformExpressionWithCache().

8.185.3.3 cache

```
template<typename Type>
exhashmap<ex> GiNaC::Gt_detail::TransformExpressionWithCache< Type >::cache  [private]
```

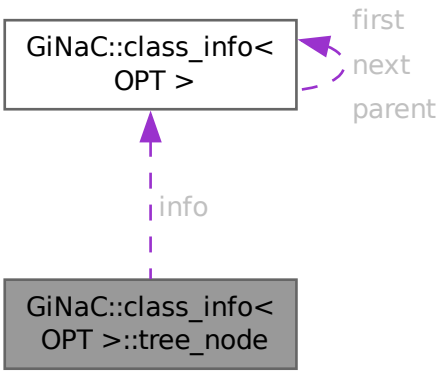
Referenced by clear_cache(), get_cache(), impl(), and print_cache().

The documentation for this class was generated from the following files:

- Gt.h
- Gt_helpers.h
- Gt_helpers.cpp

8.186 GiNaC::class_info< OPT >::tree_node Struct Reference

Collaboration diagram for GiNaC::class_info< OPT >::tree_node:



Public Member Functions

- tree_node (class_info *i)
- void add_child (tree_node *n)

Public Attributes

- std::vector< tree_node * > children
- class_info * info

8.186.1 Constructor & Destructor Documentation

8.186.1.1 tree_node()

```
template<class OPT>
GiNaC::class_info< OPT >::tree_node::tree_node (
            class_info * i)  [inline]
```

References GiNaC::class_info< OPT >::class_info(), and info.

Referenced by add_child().

8.186.2 Member Function Documentation

8.186.2.1 add_child()

```
template<class OPT>
void GiNaC::class_info< OPT >::tree_node::add_child (
            tree_node * n)  [inline]
```

References children, and tree_node().

8.186.3 Member Data Documentation

8.186.3.1 children

```
template<class OPT>
std::vector<tree_node *> GiNaC::class_info< OPT >::tree_node::children
```

Referenced by add_child(), and GiNaC::class_info< OPT >::dump_tree().

8.186.3.2 info

```
template<class OPT>
class_info* GiNaC::class_info< OPT >::tree_node::info
```

Referenced by GiNaC::class_info< OPT >::dump_tree(), and tree_node().

The documentation for this struct was generated from the following file:

- class_info.h

8.187 GiNaC::unarchive_table_t Class Reference

`#include <archive.h>`

Public Member Functions

- unarchive_table_t ()
- ~unarchive_table_t ()
- synthesize_func find (const std::string &classname) const
- void insert (const std::string &classname, synthesize_func f)

Static Private Attributes

- static int usecount = 0
- static unarchive_map_t * unarch_map = nullptr

8.187.1 Constructor & Destructor Documentation

8.187.1.1 unarchive_table_t()

```
GiNaC::unarchive_table_t::unarchive_table_t ()
```

References unarch_map, and usecount.

8.187.1.2 ~unarchive_table_t()

```
GiNaC::unarchive_table_t::~unarchive_table_t ()
```

References unarch_map, and usecount.

8.187.2 Member Function Documentation

8.187.2.1 find()

```
synthesize_func GiNaC::unarchive_table_t::find (
            const std::string & classname) const
```

References unarch_map.

Referenced by GiNaC::find_factory_fcn().

8.187.2.2 insert()

```
void GiNaC::unarchive_table_t::insert (
            const std::string & classname,
            synthesize_func f)
```

References unarch_map.

8.187.3 Member Data Documentation

8.187.3.1 usecount

```
int GiNaC::unarchive_table_t::usecount = 0  [static], [private]
```

Referenced by unarchive_table_t(), and ~unarchive_table_t().

8.187.3.2 unarch_map

```
unarchive_map_t * GiNaC::unarchive_table_t::unarch_map = nullptr  [static], [private]
```

Referenced by find(), insert(), unarchive_table_t(), and ~unarchive_table_t().

The documentation for this class was generated from the following files:

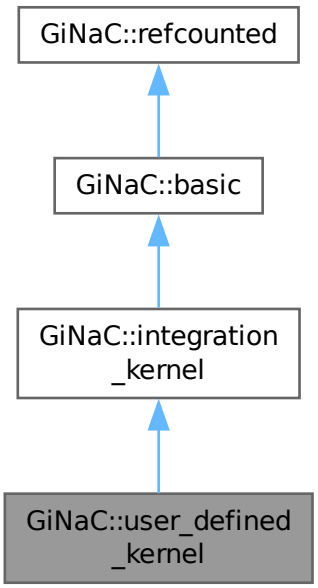
- archive.h
- archive.cpp

8.188 GiNaC::user_defined_kernel Class Reference

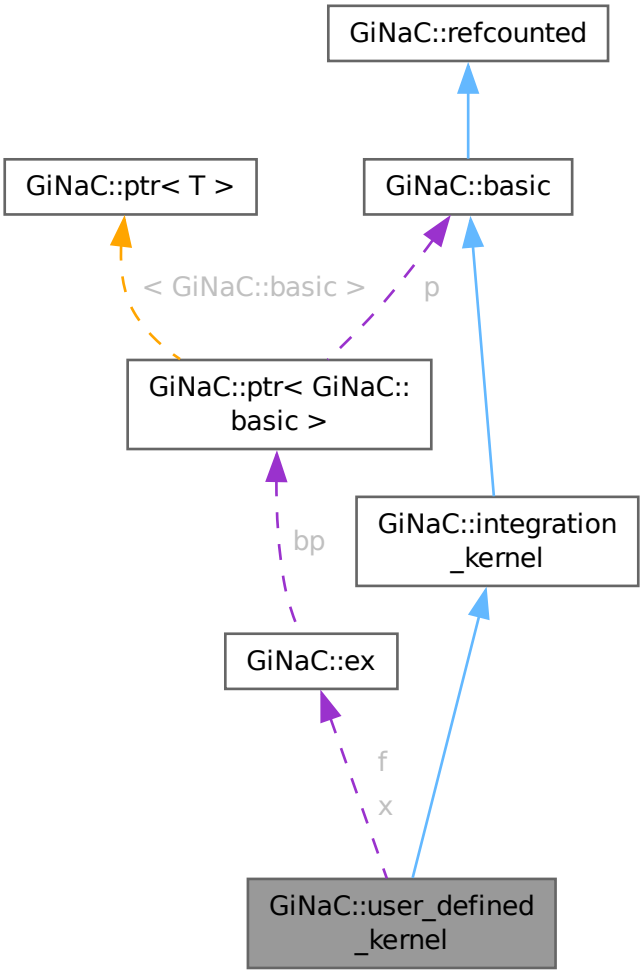
A user-defined integration kernel.

```
#include <integration_kernel.h>
```

Inheritance diagram for GiNaC::user_defined_kernel:



Collaboration diagram for GiNaC::user_defined_kernel:



Public Member Functions

- user_defined_kernel (const ex &f, const ex &x)
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex & let_op (size_t i) override

  Return modifiable operand/member at position i.
- bool is_numeric (void) const override

  This routine returns true, if the integration kernel can be evaluated numerically.
- ex Laurent_series (const ex &x, int order) const override

  Returns the Laurent series, starting possibly with the pole term.

Public Member Functions inherited from GiNaC::integration_kernel

- ex series (const relational &r, int order, unsigned options=0) const override

  Default implementation of ex::series().
- virtual bool has_trailing_zero (void) const

  This routine returns true, if the integration kernel has a trailing zero.
- virtual ex get_numerical_value (const ex &lambda, int N_trunc=0) const

  Evaluates the integrand at lambda.
- size_t get_cache_size (void) const

  Returns the current size of the cache.
- void set_cache_step (int cache_steps) const

  Sets the step size by which the cache is increased.
- ex get_series_coeff (int i) const

  Wrapper around series_coeff(i), converts cl_N to numeric.
- cln::cl_N series_coeff (int i) const

  Subclasses have either to implement series_coeff_impl or the two methods Laurent_series and uses_Laurent_series.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

Test for occurrence of a pattern.

- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- virtual void archive (archive_node &n) const

  Save (serialize) the object into archive node.
- virtual void read_archive (const archive_node &n, lst &syms)

Load (deserialize) the object from an archive node.

- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- bool uses_Laurent_series () const override

  Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::integration_kernel

- virtual cln::cl_N series_coeff_impl (int i) const

  For $\omega = d\lambda$ only the coefficient of λ^0 is non-zero.
- ex get_numerical_value_impl (const ex &lambda, const ex &pre, int shift, int N_trunc) const

  The actual implementation for computing a numerical value for the integrand.
- void do_print (const print_context &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- virtual unsigned calchash () const

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- ex f
- ex x

Protected Attributes inherited from GiNaC::integration_kernel

- int cache_step_size
- std::vector< cln::cl_N > series_vec

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.188.1 Detailed Description

A user-defined integration kernel.

The input is an expression f, depending on a variable x. It is assumed that f has a Laurent expansion around $x = 0$ and maximally a simple pole at $x = 0$.

8.188.2 Constructor & Destructor Documentation

8.188.2.1 user_defined_kernel()

```
GiNaC::user_defined_kernel::user_defined_kernel (
            const ex & f,
            const ex & x)
```

References f, user_defined_kernel(), and x.

Referenced by user_defined_kernel().

8.188.3 Member Function Documentation

8.188.3.1 nops()

```
size_t GiNaC::user_defined_kernel::nops () const  [override], [virtual]
```

Number of operands/members.

Reimplemented from GiNaC::basic.

8.188.3.2 op()

```
ex GiNaC::user_defined_kernel::op (
            size_t i) const  [override], [virtual]
```

Return operand/member at position i.

Reimplemented from GiNaC::basic.

References f, and x.

8.188.3.3 let_op()

```
ex & GiNaC::user_defined_kernel::let_op (
            size_t i)  [override], [virtual]
```

Return modifiable operand/member at position i.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ensure_if_modifiable(), f, and x.

8.188.3.4 is_numeric()

```
bool GiNaC::user_defined_kernel::is_numeric (
            void ) const  [override], [virtual]
```

This routine returns true, if the integration kernel can be evaluated numerically.

Reimplemented from GiNaC::integration_kernel.

References f, GiNaC::info_flags::numeric, GiNaC::numeric::numeric(), and x.

8.188.3.5 Laurent_series()

```
ex GiNaC::user_defined_kernel::Laurent_series (
            const ex & x,
            int order) const  [override], [virtual]
```

Returns the Laurent series, starting possibly with the pole term.

Neglected terms are of order $O(x^order)$.

Reimplemented from GiNaC::integration_kernel.

References f, GiNaC::ex::series(), GiNaC::ex::subs(), and x.

8.188.3.6 uses_Laurent_series()

```
bool GiNaC::user_defined_kernel::uses_Laurent_series () const  [override], [protected], [virtual]
```

Returns true, if the coefficients are computed from the Laurent series (in which case the method Laurent_series needs to be implemented).

Returns false if this is not the case (and the class has an implementation of series_coeff_impl).

Reimplemented from GiNaC::integration_kernel.

8.188.3.7 do_print()

```
void GiNaC::user_defined_kernel::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References f, GiNaC::ex::print(), GiNaC::print_context::s, and x.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.188.4 Member Data Documentation

8.188.4.1 f

```
ex GiNaC::user_defined_kernel::f  [protected]
```

Referenced by do_print(), is_numeric(), Laurent_series(), let_op(), op(), and user_defined_kernel().

8.188.4.2 x

```
ex GiNaC::user_defined_kernel::x  [protected]
```

Referenced by do_print(), is_numeric(), Laurent_series(), let_op(), op(), and user_defined_kernel().

The documentation for this class was generated from the following files:

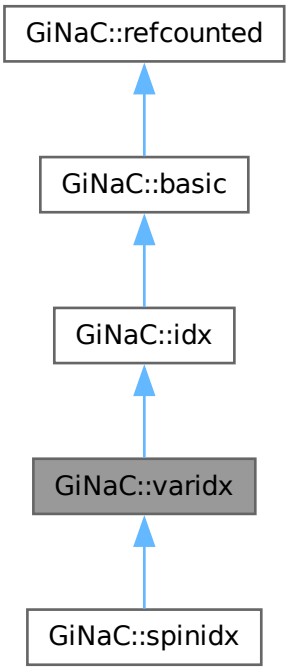
- integration_kernel.h
- integration_kernel.cpp

8.189 GiNaC::varidx Class Reference

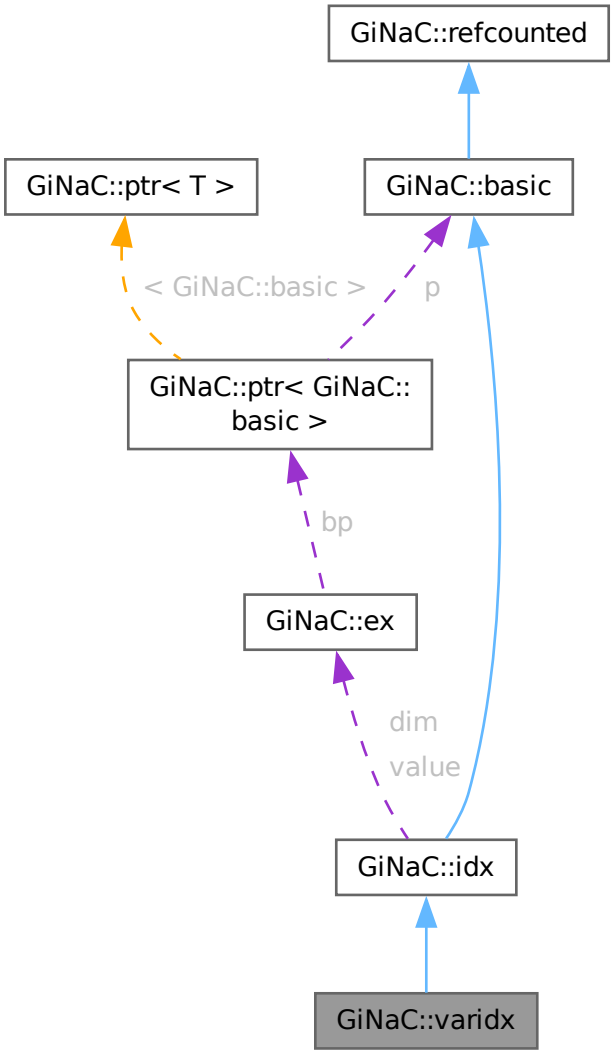
This class holds an index with a variance (co- or contravariant).

`#include <idx.h>`

Inheritance diagram for GiNaC::varidx:



Collaboration diagram for GiNaC::varidx:



Public Member Functions

- varidx (const ex &v, const ex &dim, bool covariant=false)

  Construct index with given value, dimension and variance.
- bool is_dummy_pair_same_type (const basic &other) const override

  Check whether the index forms a dummy index pair with another index of the same type.
- void archive (archive_node &n) const override

  Save (serialize) the object into archive node.
- void read_archive (const archive_node &n, lst &syms) override

  Load (deserialize) the object from an archive node.
- bool is_covariant () const

  Check whether the index is covariant.

- bool is_contravariant () const

  Check whether the index is contravariant (not covariant).
- ex toggle_variance () const

  Make a new index with the same value but the opposite variance.

Public Member Functions inherited from GiNaC::idx

- idx (const ex &v, const ex &dim)

  Construct index with given value and dimension.
- bool info (unsigned inf) const override

  Information about the object.
- size_t nops () const override

  Number of operands/members.
- ex op (size_t i) const override

  Return operand/member at position i.
- ex map (map_function &f) const override

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- ex evalf () const override

  By default, basic::evalf would evaluate the index value but we don't want a.1 to become a.
- ex subs (const exmap &m, unsigned options=0) const override

  Substitute a set of objects by arbitrary expressions.
- ex get_value () const

  Get value of index.
- bool is_numeric () const

  Check whether the index is numeric.
- bool is_symbolic () const

  Check whether the index is symbolic.
- ex get_dim () const

  Get dimension of index space.
- bool is_dim_numeric () const

  Check whether the dimension is numeric.
- bool is_dim_symbolic () const

  Check whether the dimension is symbolic.
- ex replace_dim (const ex &new_dim) const

  Make a new index with the same value but a different dimension.
- ex minimal_dim (const idx &other) const

  Return the minimum of the dimensions of this and another index.

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic.*
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic * duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.

- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

  Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.
- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual bool match (const ex &pattern, exmap &repls) const

  Check whether the expression matches a given pattern.
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

  Implementation ex::max_coefficient().
- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.

- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- bool match_same_type (const basic &other) const override

  Returns true if the attributes of two objects are similar enough for a match.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const

Protected Member Functions inherited from GiNaC::idx

- ex derivative (const symbol &s) const override

  Implementation of ex::diff() for an index always returns 0.
- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void print_index (const print_context &c, unsigned level) const
- void do_print (const print_context &c, unsigned level) const
- void do_print_csrc (const print_csrc &c, unsigned level) const
- void do_print_latex (const print_latex &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Protected Attributes

- bool covariant

  x.mu, default is contravariant: x~mu

Protected Attributes inherited from GiNaC::idx

- ex value

  Expression that constitutes the index (numeric or symbolic name).
- ex dim

  Dimension of space (can be symbolic or numeric).

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.189.1 Detailed Description

This class holds an index with a variance (co- or contravariant).

There is an associated metric tensor that can be used to raise/lower indices.

8.189.2 Constructor & Destructor Documentation

8.189.2.1 varidx()

```
GiNaC::varidx::varidx (
             const ex & v,
             const ex & dim,
             bool covariant = false)
```

Construct index with given value, dimension and variance.

Parameters

v	Value of index (numeric or symbolic)
dim	Dimension of index space (numeric or symbolic)
covariant	Make covariant index (default is contravariant)

References covariant.

Referenced by GiNaC::dirac_slash(), is_dummy_pair_same_type(), match_same_type(), and toggle_variance().

8.189.3 Member Function Documentation

8.189.3.1 is_dummy_pair_same_type()

```
bool GiNaC::varidx::is_dummy_pair_same_type (
             const basic & other) const  [override], [virtual]
```

Check whether the index forms a dummy index pair with another index of the same type.

Reimplemented from GiNaC::idx.

References covariant, and varidx().

8.189.3.2 archive()

```
void GiNaC::varidx::archive (
             archive_node & n) const  [override], [virtual]
```

Save (serialize) the object into archive node.

Archive the object.

Losely speaking, this method turns an expression into a byte stream (which can be saved and restored later on, or sent via network, etc.)

Reimplemented from GiNaC::idx.

References GiNaC::archive_node::add_bool(), and covariant.

8.189.3.3 read_archive()

```
void GiNaC::varidx::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Load (deserialize) the object from an archive node.

Construct object from archive_node.

Note

This method is essentially a constructor. However, constructors can't be virtual. So, if unarchiving routines are implemented as constructors one would need to define such a constructor in every class, even if all it does is simply calling constructor of a superclass.

Reimplemented from GiNaC::idx.

References covariant, and GiNaC::archive_node::find_bool().

8.189.3.4 match_same_type()

```
bool GiNaC::varidx::match_same_type (
            const basic & other) const  [override], [protected], [virtual]
```

Returns true if the attributes of two objects are similar enough for a match.

This function must not match subexpressions (this is already done by basic::match()). Only attributes not accessible by op() should be compared. This is also the reason why this function doesn't take the wildcard replacement list from match() as an argument: only subexpressions are subject to wildcard matches. Also, this function only needs to be implemented for container classes because is_equal_same_type() is automatically used instead of match_same_type() if nops() == 0.

See also

basic::match

Reimplemented from GiNaC::idx.

References covariant, GINAC_ASSERT, GiNaC::is_a(), and varidx().

8.189.3.5 is_covariant()

```
bool GiNaC::varidx::is_covariant () const  [inline]
```

Check whether the index is covariant.

References covariant.

Referenced by GiNaC::spinmetric::contract_with(), and GiNaC::tensmetric::eval_indexed().

8.189.3.6 is_contravariant()

```
bool GiNaC::varidx::is_contravariant () const  [inline]
```

Check whether the index is contravariant (not covariant).

References covariant.

8.189.3.7 toggle_variance()

```
ex GiNaC::varidx::toggle_variance () const
```

Make a new index with the same value but the opposite variance.

References GiNaC::basic::clearflag(), covariant, GiNaC::basic::duplicate(), GiNaC::status_flags::hash_calculated, and varidx().

Referenced by GiNaC::base_and_index(), and GiNaC::clifford::eval_ncmul().

8.189.3.8 do_print()

```
void GiNaC::varidx::do_print (
             const print_context & c,
             unsigned level) const  [protected]
```

References covariant, GiNaC::idx::print_index(), and GiNaC::print_context::s.

8.189.3.9 do_print_tree()

```
void GiNaC::varidx::do_print_tree (
             const print_tree & c,
             unsigned level) const  [protected]
```

References covariant, GiNaC::print_tree::delta_indent, GiNaC::idx::dim, GiNaC::basic::flags, GiNaC::basic::hashvalue, GiNaC::ex::print(), GiNaC::print_context::s, and GiNaC::idx::value.

8.189.4 Member Data Documentation

8.189.4.1 covariant

```
bool GiNaC::varidx::covariant  [protected]
```

x.mu, default is contravariant: x∼mu

Referenced by archive(), GiNaC::spinidx::do_print(), do_print(), GiNaC::spinidx::do_print_tree(), do_print_tree(), is_contravariant(), is_covariant(), is_dummy_pair_same_type(), match_same_type(), read_archive(), toggle_variance(), GiNaC::spinidx::toggle_variance_dot(), and varidx().

The documentation for this class was generated from the following files:

- idx.h
- idx.cpp

8.190 GiNaC::visitor Class Reference

Degenerate base class for visitors.

`#include <basic.h>`

Protected Member Functions

- virtual ~visitor ()

8.190.1 Detailed Description

Degenerate base class for visitors.

basic and derivative classes support Robert C. Martin's Acyclic Visitor pattern (cf. http://condor.depaul.edu/dmumaugh/OOT/Design-Principles/acv.pdf or chapter 10 of Andrei Alexandrescu's "Modern C++ Design").

8.190.2 Constructor & Destructor Documentation

8.190.2.1 ~visitor()

`virtual GiNaC::visitor::~visitor () [inline], [protected], [virtual]`

The documentation for this class was generated from the following file:

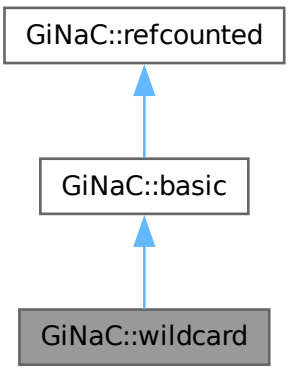
- basic.h

8.191 GiNaC::wildcard Class Reference

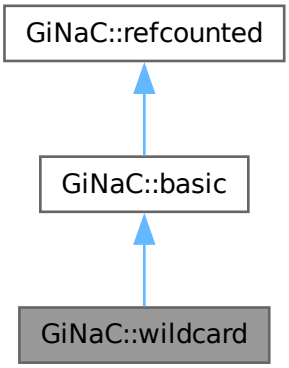
This class acts as a wildcard for subs(), match(), has() and find().

`#include <wildcard.h>`

Inheritance diagram for GiNaC::wildcard:



Collaboration diagram for GiNaC::wildcard:



Public Member Functions

- wildcard (unsigned label)

  Construct wildcard with specified label.
- bool match (const ex &pattern, exmap &repl_lst) const override

  Check whether the expression matches a given pattern.
- void archive (archive_node &n) const override

  Save (a.k.a.
- void read_archive (const archive_node &n, lst &syms) override

  Read (a.k.a.
- unsigned get_label () const

Public Member Functions inherited from GiNaC::basic

- virtual ~basic ()

  basic destructor, virtual because class ex will delete objects of derived classes via a basic∗.
- basic (const basic &other)
- const basic & operator= (const basic &other)

  basic assignment operator: the other object might be of a derived class.
- virtual basic ∗ duplicate () const

  Create a clone of this object on the heap.
- virtual ex eval () const

  Perform automatic non-interruptive term rewriting rules.
- virtual ex evalf () const

  Evaluate object numerically.
- virtual ex evalm () const

  Evaluate sums, products and integer powers of matrices.
- virtual ex eval_integ () const

  Evaluate integrals, if result is known.
- virtual ex eval_indexed (const basic &i) const

Perform automatic symbolic evaluations on indexed expression that contains this object as the base expression.

- virtual void print (const print_context &c, unsigned level=0) const

  Output to stream.
- virtual void dbgprint () const

  Little wrapper around print to be called within a debugger.
- virtual void dbgprinttree () const

  Little wrapper around printtree to be called within a debugger.
- virtual unsigned precedence () const

  Return relative operator precedence (for parenthezing output).
- virtual bool info (unsigned inf) const

  Information about the object.
- virtual size_t nops () const

  Number of operands/members.
- virtual ex op (size_t i) const

  Return operand/member at position i.
- virtual ex operator[ ] (const ex &index) const
- virtual ex operator[ ] (size_t i) const
- virtual ex & let_op (size_t i)

  Return modifiable operand/member at position i.
- virtual ex & operator[ ] (const ex &index)
- virtual ex & operator[ ] (size_t i)
- virtual bool has (const ex &other, unsigned options=0) const

  Test for occurrence of a pattern.
- virtual ex subs (const exmap &m, unsigned options=0) const

  Substitute a set of objects by arbitrary expressions.
- virtual ex map (map_function &f) const

  Construct new expression by applying the specified function to all sub-expressions (one level only, not recursively).
- virtual void accept (GiNaC::visitor &v) const
- virtual bool is_polynomial (const ex &var) const

  Check whether this is a polynomial in the given variables.
- virtual int degree (const ex &s) const

  Return degree of highest power in object s.
- virtual int ldegree (const ex &s) const

  Return degree of lowest power in object s.
- virtual ex coeff (const ex &s, int n=1) const

  Return coefficient of degree n in object s.
- virtual ex expand (unsigned options=0) const

  Expand expression, i.e.
- virtual ex collect (const ex &s, bool distributed=false) const

  Sort expanded expression in terms of powers of some object(s).
- virtual ex series (const relational &r, int order, unsigned options=0) const

  Default implementation of ex::series().
- virtual ex normal (exmap &repl, exmap &rev_lookup, lst &modifier) const

  Default implementation of ex::normal().
- virtual ex to_rational (exmap &repl) const

  Default implementation of ex::to_rational().
- virtual ex to_polynomial (exmap &repl) const
- virtual numeric integer_content () const
- virtual ex smod (const numeric &xi) const

  Apply symmetric modular homomorphism to an expanded multivariate polynomial.
- virtual numeric max_coefficient () const

Implementation ex::max_coefficient().

- virtual exvector get_free_indices () const

  Return a vector containing the free indices of an expression.
- virtual ex add_indexed (const ex &self, const ex &other) const

  Add two indexed expressions.
- virtual ex scalar_mul_indexed (const ex &self, const numeric &other) const

  Multiply an indexed expression with a scalar.
- virtual bool contract_with (exvector::iterator self, exvector::iterator other, exvector &v) const

  Try to contract two indexed expressions that appear in the same product.
- virtual unsigned return_type () const
- virtual return_type_t return_type_tinfo () const
- virtual ex conjugate () const
- virtual ex real_part () const
- virtual ex imag_part () const
- template<class T>
  void print_dispatch (const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- void print_dispatch (const registered_class_info &ri, const print_context &c, unsigned level) const

  Like print(), but dispatch to the specified class.
- ex subs_one_level (const exmap &m, unsigned options) const

  Helper function for subs().
- ex diff (const symbol &s, unsigned nth=1) const

  Default interface of nth derivative ex::diff(s, n).
- int compare (const basic &other) const

  Compare objects syntactically to establish canonical ordering.
- bool is_equal (const basic &other) const

  Test for syntactic equality.
- const basic & hold () const

  Stop further evaluation.
- unsigned gethash () const
- const basic & setflag (unsigned f) const

  Set some status_flags.
- const basic & clearflag (unsigned f) const

  Clear some status_flags.

Public Member Functions inherited from GiNaC::refcounted

- refcounted () noexcept
- unsigned int add_reference () noexcept
- unsigned int remove_reference () noexcept
- unsigned int get_refcount () const noexcept
- void set_refcount (unsigned int r) noexcept

Protected Member Functions

- unsigned calchash () const override

  Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.
- void do_print (const print_context &c, unsigned level) const
- void do_print_tree (const print_tree &c, unsigned level) const
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

Protected Member Functions inherited from GiNaC::basic

- basic ()
- virtual ex eval_ncmul (const exvector &v) const
- virtual bool match_same_type (const basic &other) const

  Returns true if the attributes of two objects are similar enough for a match.
- virtual ex derivative (const symbol &s) const

  Default implementation of ex::diff().
- virtual int compare_same_type (const basic &other) const

  Returns order relation between two objects of same type.
- virtual bool is_equal_same_type (const basic &other) const

  Returns true if two objects of same type are equal.
- void ensure_if_modifiable () const

  Ensure the object may be modified without hurting others, throws if this is not the case.
- void do_print (const print_context &c, unsigned level) const

  Default output to stream.
- void do_print_tree (const print_tree &c, unsigned level) const

  Tree output to stream.
- void do_print_python_repr (const print_python_repr &c, unsigned level) const

  Python parsable output to stream.

Private Attributes

- unsigned label

  Label used to distinguish different wildcards.

Additional Inherited Members

Protected Attributes inherited from GiNaC::basic

- unsigned flags

  of type status_flags
- unsigned hashvalue

  hash value

8.191.1 Detailed Description

This class acts as a wildcard for subs(), match(), has() and find().

An integer label is used to identify different wildcards.

8.191.2 Constructor & Destructor Documentation

8.191.2.1 wildcard()

```
GiNaC::wildcard::wildcard (
            unsigned label)
```

Construct wildcard with specified label.

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, label, GiNaC::basic::setflag(), and wildcard().

Referenced by GiNaC::wild(), and wildcard().

8.191.3 Member Function Documentation

8.191.3.1 match()

```
bool GiNaC::wildcard::match (
            const ex & pattern,
            exmap & repl_lst) const  [override], [virtual]
```

Check whether the expression matches a given pattern.

For every wildcard object in the pattern, a pair with the wildcard as a key and matching expression as a value is added to repl_lst.

Reimplemented from GiNaC::basic.

References GiNaC::basic::ex, GiNaC::ex_to(), and GiNaC::basic::is_equal().

8.191.3.2 archive()

```
void GiNaC::wildcard::archive (
            archive_node & n) const  [override], [virtual]
```

Save (a.k.a.

serialize) object into archive.

Reimplemented from GiNaC::basic.

References GiNaC::archive_node::add_unsigned(), and label.

8.191.3.3 read_archive()

```
void GiNaC::wildcard::read_archive (
            const archive_node & n,
            lst & syms)  [override], [virtual]
```

Read (a.k.a.

deserialize) object from archive.

Reimplemented from GiNaC::basic.

References GiNaC::status_flags::evaluated, GiNaC::status_flags::expanded, GiNaC::archive_node::find_unsigned(), label, and GiNaC::basic::setflag().

8.191.3.4 calchash()

```
unsigned GiNaC::wildcard::calchash () const  [override], [protected], [virtual]
```

Compute the hash value of an object and if it makes sense to store it in the objects status_flags, do so.

The method inherited from class basic computes a hash value based on the type and hash values of possible members. For this reason it is well suited for container classes but atomic classes should override this implementation because otherwise they would all end up with the same hashvalue.

Reimplemented from GiNaC::basic.

References GiNaC::golden_ratio_hash(), GiNaC::status_flags::hash_calculated, GiNaC::basic::hashvalue, label, and GiNaC::basic::setflag().

8.191.3.5 get_label()

```
unsigned GiNaC::wildcard::get_label () const  [inline]
```

References label.

8.191.3.6 do_print()

```
void GiNaC::wildcard::do_print (
            const print_context & c,
            unsigned level) const  [protected]
```

References label, and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.191.3.7 do_print_tree()

```
void GiNaC::wildcard::do_print_tree (
            const print_tree & c,
            unsigned level) const  [protected]
```

References GiNaC::basic::flags, GiNaC::basic::hashvalue, label, and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.191.3.8 do_print_python_repr()

```
void GiNaC::wildcard::do_print_python_repr (
            const print_python_repr & c,
            unsigned level) const  [protected]
```

References label, and GiNaC::print_context::s.

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

8.191.4 Member Data Documentation

8.191.4.1 label

```
unsigned GiNaC::wildcard::label  [private]
```

Label used to distinguish different wildcards.

Referenced by archive(), calchash(), do_print(), do_print_python_repr(), do_print_tree(), get_label(), read_archive(), and wildcard().

The documentation for this class was generated from the following files:

- wildcard.h
- wildcard.cpp

8.192 GiNaC::zeta1_SERIAL Class Reference

Complex conjugate.

`#include <inifcns.h>`

Static Public Attributes

- static unsigned serial

8.192.1 Detailed Description

Complex conjugate.

Real part. Imaginary part. Absolute value. Step function. Complex sign. Eta function: log(a∗b) == log(a) + log(b) + eta(a, b). Sine. Cosine. Tangent. Exponential function. Natural logarithm. Inverse sine (arc sine). Inverse cosine (arc cosine). Inverse tangent (arc tangent). Inverse tangent with two arguments. Hyperbolic Sine. Hyperbolic Cosine. Hyperbolic Tangent. Inverse hyperbolic Sine (area hyperbolic sine). Inverse hyperbolic Cosine (area hyperbolic cosine). Inverse hyperbolic Tangent (area hyperbolic tangent). Dilogarithm. Trilogarithm. Derivatives of Riemann's Zeta-function. Multiple zeta value including Riemann's zeta-function.

8.192.2 Member Data Documentation

8.192.2.1 serial

`unsigned GiNaC::zeta1_SERIAL::serial [static]`

Initial value:

```
= function::register_new(function_options("zeta", 1).
  evalf_func(zeta1_evalf).
  eval_func(zeta1_eval).
  derivative_func(zeta1_deriv).
  print_func<print_latex>(zeta1_print_latex).
  do_not_evalf_params().
  overloaded(2))
```

Referenced by GiNaC::zeta().

The documentation for this class was generated from the following files:

- inifcns.h
- inifcns_nstdsums.cpp

8.193 GiNaC::zeta2_SERIAL Class Reference

Alternating Euler sum or colored MZV.

`#include <inifcns.h>`

Static Public Attributes

- static unsigned serial

8.193.1 Detailed Description

Alternating Euler sum or colored MZV.

8.193.2 Member Data Documentation

8.193.2.1 serial

```
unsigned GiNaC::zeta2_SERIAL::serial  [static]
```

Initial value:

```
= function::register_new(function_options("zeta", 2).
  evalf_func(zeta2_evalf).
  eval_func(zeta2_eval).
  derivative_func(zeta2_deriv).
  print_func<print_latex>(zeta2_print_latex).
  do_not_evalf_params().
  overloaded(2))
```

Referenced by GiNaC::zeta().

The documentation for this class was generated from the following files:

- inifcns.h
- inifcns_nstdsums.cpp

Chapter 9

File Documentation

9.1 add.cpp File Reference

Implementation of GiNaC's sums of expressions.

```
#include "add.h"
#include "mul.h"
#include "archive.h"
#include "operators.h"
#include "matrix.h"
#include "utils.h"
#include "clifford.h"
#include "ncmul.h"
#include "compiler.h"
#include <limits>
#include <stdexcept>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (add, expairseq, print_func< print_context >(&add::do_print). print_func< print_latex >(&add::do_print_latex). print_func< print_csrc >(&add::do_print_csrc). print_func< print_tree >(&add::do_print_tree). print_func< print_python_repr >(&add::do_print_python_repr)) add
- GiNaC::GINAC_BIND_UNARCHIVER (add)

9.1.1 Detailed Description

Implementation of GiNaC's sums of expressions.

9.2 add.h File Reference

Interface to GiNaC's sums of expressions.

```
#include "expairseq.h"
```

Classes

- class GiNaC::add
  Sum of expressions.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (add)

9.2.1 Detailed Description

Interface to GiNaC's sums of expressions.

9.3 archive.cpp File Reference

Archiving of GiNaC expressions.

```
#include "archive.h"
#include "registrar.h"
#include "ex.h"
#include "lst.h"
#include "version.h"
#include <stdexcept>
```

Namespaces

- namespace GiNaC

Functions

- static void GiNaC::write_unsigned (std::ostream &os, unsigned val)

  Write unsigned integer quantity to stream.
- static unsigned GiNaC::read_unsigned (std::istream &is)

  Read unsigned integer quantity from stream.
- std::ostream & GiNaC::operator<< (std::ostream &os, const archive_node &n)

  Write archive_node to binary data stream.
- std::ostream & GiNaC::operator<< (std::ostream &os, const archive &ar)

  Write archive to binary data stream.
- std::istream & GiNaC::operator>> (std::istream &is, archive_node &n)

  Read archive_node from binary data stream.
- std::istream & GiNaC::operator>> (std::istream &is, archive &ar)

  Read archive from binary data stream.
- static synthesize_func GiNaC::find_factory_fcn (const std::string &name)

9.3.1 Detailed Description

Archiving of GiNaC expressions.

9.4 archive.h File Reference

Archiving of GiNaC expressions.

```
#include "ex.h"
#include <iosfwd>
#include <map>
#include <string>
#include <vector>
```

Classes

- class GiNaC::archive_node

  This class stores all properties needed to record/retrieve the state of one object of class basic (or a derived class).
- struct GiNaC::archive_node::property_info

  Information about a stored property.
- struct GiNaC::archive_node::property

  Archived property (data type, name and associated data).
- struct GiNaC::archive_node::archive_node_cit_range
- class GiNaC::unarchive_table_t
- class GiNaC::archive

  This class holds archived versions of GiNaC expressions (class ex).
- struct GiNaC::archive::archived_ex

  Archived expression descriptor.

Namespaces

- namespace GiNaC

Macros

- #define GINAC_DECLARE_UNARCHIVER(classname)

  Helper macros to register a class with (un)archiving (a.k.a.
- #define GINAC_BIND_UNARCHIVER(classname)

Typedefs

- typedef unsigned GiNaC::archive_node_id

  Numerical ID value to refer to an archive_node.
- typedef unsigned GiNaC::archive_atom

  Numerical ID value to refer to a string.
- typedef basic ∗(∗) GiNaC::synthesize_func()
- typedef std::map< std::string, synthesize_func > GiNaC::unarchive_map_t

Functions

- std::ostream & GiNaC::operator<< (std::ostream &os, const archive &ar)

  Write archive to binary data stream.
- std::istream & GiNaC::operator>> (std::istream &is, archive &ar)

  Read archive from binary data stream.

Variables

- static unarchive_table_t GiNaC::unarch_table_instance

9.4.1 Detailed Description

Archiving of GiNaC expressions.

9.4.2 Macro Definition Documentation

9.4.2.1 GINAC_DECLARE_UNARCHIVER

```
#define GINAC_DECLARE_UNARCHIVER(
              classname)
```

Value:

```
class classname ## _unarchiver                  \
{                                 \
    static int usecount;                    \
public:                                 \
    static GiNaC::basic* create();              \
    classname ## _unarchiver();             \
    ~ classname ## _unarchiver();               \
};                                \
static classname ## _unarchiver classname ## _unarchiver_instance
```

Helper macros to register a class with (un)archiving (a.k.a.

(de)serialization).

Usage: put

GINAC_DECLARE_UNARCHIVER(myclass);

into the header file (in the global or namespace scope), and

GINAC_BIND_UNARCHIVER(myclass);

into the source file.

Effect: the `myclass' (being a class derived directly or indirectly from GiNaC::basic) can be archived and unarchived.

Note: you need to use GINAC_{DECLARE,BIND}_UNARCHIVER incantations in order to make your class (un)archivable *even if your class does not overload `read_archive' method*. Sorry for inconvenience.

How it works:

The `basic' class has a read_archive' virtual method which reads an expression from archive. Derived classes can overload that method. There's a small problem, though. On unarchiving all we have is a set of named byte streams. In C++ the class name (as written in the source code) has nothing to do with its actual type. Thus, we need establish a correspondence ourselves. To do so we maintain a `class_name' => function_pointer' table (see the unarchive_table_t class above). Every function in this table is supposed to create a new object of the `class_name' type. The archive_node' class uses that table to construct an object of correct type. Next it invokes read_archive virtual method of newly created object, which does the actual job.

Note: this approach is very simple-minded (it does not handle classes with same names from different namespaces, multiple inheritance, etc), but it happens to work surprisingly well.

9.4.2.2 GINAC_BIND_UNARCHIVER

```
#define GINAC_BIND_UNARCHIVER(
                classname)
```

Value:

```
classname ## _unarchiver::classname ## _unarchiver()        \
{                                                            \
    static GiNaC::unarchive_table_t table;                   \
    if (usecount++ == 0) {                                   \
        table.insert(std::string(#classname),                \
            &(classname ## _unarchiver::create));            \
    }                                                        \
}                                                            \
GiNaC::basic* classname ## _unarchiver::create()             \
{                                                            \
    return new classname();                                  \
}                                                            \
classname ## _unarchiver::~ classname ## _unarchiver() { }   \
int classname ## _unarchiver::usecount = 0
```

9.5 assertion.h File Reference

Assertion macro definition.

Macros

- #define GINAC_ASSERT(X)

  Assertion macro for checking invariances.

9.5.1 Detailed Description

Assertion macro definition.

9.5.2 Macro Definition Documentation

9.5.2.1 GINAC_ASSERT

```
#define GINAC_ASSERT(
              X)
```

Value:

```
((void)0)
```

Assertion macro for checking invariances.

Referenced by GiNaC::acos_deriv(), GiNaC::acosh_deriv(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::add::add(), GiNaC::symmetry::add(), GiNaC::remember_table::add_entry(), GiNaC::remember_table_list::add_entry(), GiNaC::matrix::add_indexed(), GiNaC::algebraic_match_mul_with_mul(), GiNaC::function::archive(), GiNaC::asin_deriv(), GiNaC::asinh_deriv(), GiNaC::atan2_deriv(), GiNaC::atan_deriv(), GiNaC::atan_series(), GiNaC::atanh_deriv(), GiNaC::atanh_series(), GiNaC::base_and_index(), GiNaC::beta_deriv(), GiNaC::beta_series(), GiNaC::mul::can_make_flat(), GiNaC::symmetry::canonicalize, GiNaC::clifford::clifford(), GiNaC::clifford_to_lst(), GiNaC::pseries::coeff(), GiNaC::add::combine_ex_with_coeff_to_pair(), GiNaC::expairseq::combine_ex_with_
GiNaC::mul::combine_ex_with_coeff_to_pair(), GiNaC::expairseq::combine_overall_coeff(), GiNaC::expairseq::combine_overall_coef
GiNaC::mul::combine_overall_coeff(), GiNaC::mul::combine_overall_coeff(), GiNaC::add::combine_pair_with_coeff_to_pair(), GiNaC::expairseq::combine_pair_with_coeff_to_pair(), GiNaC::mul::combine_pair_with_coeff_to_pair(), GiNaC::function::conjugate(), GiNaC::ex::construct_from_basic(), GiNaC::cliffordunit::contract_with(), GiNaC::diracgamma::contract_with(), GiNaC::matrix::contract_with(), GiNaC::spinmetric::contract_with(), GiNaC::su3d::contract_with(), GiNaC::su3f::contract_with(), GiNaC::su3t::contract_with(), GiNaC::tensdelta::contract_with(), GiNaC::tensepsilon::contract_with(), GiNaC::tensmetric::contract_wit
GiNaC::cos_deriv(), GiNaC::cosh_deriv(), GiNaC::ex::denom(), GiNaC::matrix::determinant(), GiNaC::matrix::division_free_eliminatio
GiNaC::mul::do_print_latex(), GiNaC::add::eval(), GiNaC::function::eval(), GiNaC::indexed::eval(), GiNaC::mul::eval(), GiNaC::ncmul::eval(), GiNaC::power::eval(), GiNaC::matrix::eval_indexed(), GiNaC::minkmetric::eval_indexed(), GiNaC::spinmetric::eval_indexed(), GiNaC::su3d::eval_indexed(), GiNaC::su3f::eval_indexed(), GiNaC::tensdelta::eval_indexed(), GiNaC::tensepsilon::eval_indexed(), GiNaC::tensmetric::eval_indexed(), GiNaC::function::evalf(), GiNaC::ex::ex(), GiNaC::ex::ex(), GiNaC::ex::ex(), GiNaC::ex::ex(), GiNaC::ex::ex(), GiNaC::ex::ex(), GiNaC::ex::ex(), GiNaC::ex::ex(), GiNaC::ex::ex(), GiNaC::ex::ex_to, GiNaC::exp_deriv(), GiNaC::expair::expair(), GiNaC::expairseq::expairseq(), GiNaC::expairseq::expairseq(), GiNaC::expairseq::expairseq(), GiNaC::expairseq::expairseq(), GiNaC::function::expand(), GiNaC::indexed::expand(), GiNaC::power::expand_add(), GiNaC::power::expand_add_2(), GiNaC::power::expand_mul(), GiNaC::function::expl_derivative(), GiNaC::frac_cancel(), GiNaC::matrix::fraction_free_elimination(), GiNaC::matrix::gauss_eliminatio
GiNaC::gcd_pf_mul(), GiNaC::gcd_pf_pow(), GiNaC::indexed::get_indices(), GiNaC::function::get_name(), GiNaC::GINAC_BIND_UNARCHIVER(), GiNaC::H_deriv(), GiNaC::function::imag_part(), GiNaC::function::info(), GiNaC::add::integer_content(), GiNaC::mul::integer_content(), GiNaC::expair::is_canonical_numeric(), GiNaC::remember_table_entry
GiNaC::constant::is_equal_same_type(), GiNaC::container< C >::is_equal_same_type(), GiNaC::fderivative::is_equal_same_type(), GiNaC::function::is_equal_same_type(), GiNaC::numeric::is_equal_same_type(), GiNaC::structure< T, ComparisonPolicy >::is_equa
GiNaC::symbol::is_equal_same_type(), GiNaC::clifford::let_op(), GiNaC::container< C >::let_op(), GiNaC::matrix::let_op(), GiNaC::lgamma_deriv(), GiNaC::Li2_deriv(), GiNaC::Li_deriv(), GiNaC::Li_eval(), GiNaC::log_deriv(), GiNaC::log_series(), GiNaC::remember_table::lookup_entry(), GiNaC::lsolve(), GiNaC::ex::makewriteable(), GiNaC::matrix::markowitz_elimination(), GiNaC::clifford::match_same_type(), GiNaC::color::match_same_type(), GiNaC::fderivative::match_same_type(), GiNaC::function::match_same_type(), GiNaC::idx::match_same_type(), GiNaC::matrix::match_same_type(), GiNaC::relational::match_same_type(), GiNaC::spinidx::match_same_type(), GiNaC::varidx::match_same_type(), GiNaC::add::max_coefficient(), GiNaC::mul::max_coefficient(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::mul::mul(), GiNaC::add::normal(), GiNaC::ex::normal(), GiNaC::ex::numer(), GiNaC::ex::numer_denom(), GiNaC::clifford::op(), GiNaC::container< C >::op(), GiNaC::expairseq::op(), GiNaC::idx::op(), GiNaC::integral::op(), GiNaC::matrix::op(), GiNaC::power::op(), GiNaC::relational::op(), GiNaC::sy_is_less::operator()(), GiNaC::sy_swap::operator()(), GiNaC::Order_series(), GiNaC::function::pderivative(),

GiNaC::permute_free_index_to_front(), GiNaC::matrix::pivot(), GiNaC::function::power(), GiNaC::function::print(), GiNaC::product_to_exvector(), GiNaC::pseries::pseries(), GiNaC::pseries::pseries(), GiNaC::psi1_deriv(), GiNaC::psi2_deriv(), GiNaC::ptr< T >::ptr(), GiNaC::matrix::rank(), GiNaC::expairseq::read_archive(), GiNaC::function::real_part(), GiNaC::rename_dummy_indices(), GiNaC::function::return_type(), GiNaC::ncmul::return_type(), GiNaC::relational::return_type(), GiNaC::function::return_type_tinfo(), GiNaC::relational::return_type_tinfo(), GiNaC::S_deriv(), GiNaC::matrix::scalar_mul_indexed(), GiNaC::function::series(), GiNaC::mul::series(), GiNaC::pseries::series(), GiNaC::symbol::series(), GiNaC::indexed::simplify_indexed(), GiNaC::sin_deriv(), GiNaC::sinh_deriv(), GiNaC::add::smod(), GiNaC::mul::smod(), GiNaC::sqrfree_parfrac(), GiNaC::ex::subs(), GiNaC::ex::subs(), GiNaC::ex::swap(), GiNaC::tan_deriv(), GiNaC::tan_series(), GiNaC::tanh_deriv(), GiNaC::tanh_series(), GiNaC::tgamma_deriv(), GiNaC::numeric::to_double(), GiNaC::numeric::to_int(), GiNaC::numeric::to_long(), GiNaC::indexed::validate(), GiNaC::zeta1_deriv(), GiNaC::zeta2_deriv(), GiNaC::zetaderiv_deriv(), and GiNaC::basic::~basic().

9.6 basic.cpp File Reference

Implementation of GiNaC's ABC.

```
#include "basic.h"
#include "ex.h"
#include "numeric.h"
#include "power.h"
#include "add.h"
#include "symbol.h"
#include "lst.h"
#include "ncmul.h"
#include "relational.h"
#include "operators.h"
#include "wildcard.h"
#include "archive.h"
#include "utils.h"
#include "hash_seed.h"
#include "inifcns.h"
#include <iostream>
#include <stdexcept>
#include <typeinfo>
```

Classes

- struct GiNaC::evalf_map_function

  Function object to be applied by basic::evalf().
- struct GiNaC::evalm_map_function

  Function object to be applied by basic::evalm().
- struct GiNaC::eval_integ_map_function

  Function object to be applied by basic::eval_integ().
- struct GiNaC::derivative_map_function

  Function object to be applied by basic::derivative().
- struct GiNaC::expand_map_function

  Function object to be applied by basic::expand().

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (basic, void, print_func< print_context >(&basic::do_print). print_func< print_tree >(&basic::do_print_tree). print_func< print_python_repr >(&basic::do_print_python_repr)) basic

  basic copy constructor: implicitly assumes that the other class is of the exact same type (as it's used by duplicate()), so it can copy the tinfo_key and the hash value.

Variables

- GiNaC::evalm_map_function GiNaC::map_evalm
- GiNaC::eval_integ_map_function GiNaC::map_eval_integ

9.6.1 Detailed Description

Implementation of GiNaC's ABC.

9.7 basic.h File Reference

Interface to GiNaC's ABC.

```
#include "flags.h"
#include "ptr.h"
#include "assertion.h"
#include "registrar.h"
#include <cstddef>
#include <map>
#include <set>
#include <typeinfo>
#include <vector>
#include <utility>
```

Classes

- struct GiNaC::map_function

  Function object for map().
- class GiNaC::visitor

  Degenerate base class for visitors.
- class GiNaC::basic

  This class is the ABC (abstract base class) of GiNaC's class hierarchy.

Namespaces

- namespace GiNaC

Typedefs

- typedef std::vector< ex > GiNaC::exvector
- typedef std::set< ex, ex_is_less > GiNaC::exset
- typedef std::map< ex, ex, ex_is_less > GiNaC::exmap

Functions

- template<class T>
  bool GiNaC::is_a (const basic &obj)

  Check if obj is a T, including base classes.
- template<class T>
  bool GiNaC::is_exactly_a (const basic &obj)

  Check if obj is a T, not including base classes.
- template<class B, typename... Args>
  B & GiNaC::dynallocate (Args &&... args)

  Constructs a new (class basic or derived) B object on the heap.
- template<class B>
  B & GiNaC::dynallocate (std::initializer_list< ex > il)

  Constructs a new (class basic or derived) B object on the heap.

9.7.1 Detailed Description

Interface to GiNaC's ABC.

9.8 class_info.h File Reference

Helper templates to provide per-class information for class hierarchies.

```
#include <cstddef>
#include <cstring>
#include <iomanip>
#include <iostream>
#include <map>
#include <stdexcept>
#include <string>
#include <vector>
```

Classes

- class GiNaC::class_info< OPT >
- struct GiNaC::class_info< OPT >::tree_node

Namespaces

- namespace GiNaC

Variables

- template<class OPT>
  class_info< OPT > * GiNaC::class_info< OPT >::first = nullptr
- template<class OPT>
  bool GiNaC::class_info< OPT >::parents_identified = false

9.8.1 Detailed Description

Helper templates to provide per-class information for class hierarchies.

9.9 clifford.cpp File Reference

Implementation of GiNaC's clifford algebra (Dirac gamma) objects.

```
#include "clifford.h"
#include "ex.h"
#include "idx.h"
#include "ncmul.h"
#include "symbol.h"
#include "numeric.h"
#include "symmetry.h"
#include "lst.h"
#include "relational.h"
#include "operators.h"
#include "add.h"
#include "mul.h"
#include "power.h"
#include "matrix.h"
#include "archive.h"
#include "utils.h"
#include <algorithm>
#include <stdexcept>
```

Classes

- struct GiNaC::is_not_a_clifford
  Predicate for finding non-clifford objects.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (clifford, indexed, print_func< print_dflt >(&clifford::do_print_dflt). print_func< print_latex >(&clifford::do_print_latex). print_func< print_tree >(&clifford::do_print_tree)) GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(diracone
- GiNaC::print_func< print_dflt > (&diracone::do_print). print_func< print_latex >(&diracone
- GiNaC::GINAC_BIND_UNARCHIVER (clifford)
- GiNaC::GINAC_BIND_UNARCHIVER (cliffordunit)
- GiNaC::GINAC_BIND_UNARCHIVER (diracone)
- GiNaC::GINAC_BIND_UNARCHIVER (diracgamma)
- GiNaC::GINAC_BIND_UNARCHIVER (diracgamma5)
- GiNaC::GINAC_BIND_UNARCHIVER (diracgammaL)
- GiNaC::GINAC_BIND_UNARCHIVER (diracgammaR)
- static bool GiNaC::is_dirac_slash (const ex &seq0)
- static void GiNaC::base_and_index (const ex &c, ex &b, ex &i)

  This function decomposes gamma~mu -> (1, mu) and a\ -> (a.ix, ix).
- ex GiNaC::dirac_ONE (unsigned char rl=0)

  Create a Clifford unity object.
- static unsigned GiNaC::get_dim_uint (const ex &e)
- ex GiNaC::clifford_unit (const ex &mu, const ex &metr, unsigned char rl=0)

  Create a Clifford unit object.
- ex GiNaC::dirac_gamma (const ex &mu, unsigned char rl=0)

  Create a Dirac gamma object.
- ex GiNaC::dirac_gamma5 (unsigned char rl=0)

  Create a Dirac gamma5 object.
- ex GiNaC::dirac_gammaL (unsigned char rl=0)

  Create a Dirac gammaL object.
- ex GiNaC::dirac_gammaR (unsigned char rl=0)

  Create a Dirac gammaR object.
- ex GiNaC::dirac_slash (const ex &e, const ex &dim, unsigned char rl=0)

  *Create a term of the form e_mu * gamma~mu with a unique index mu.*
- static unsigned char GiNaC::get_representation_label (const return_type_t &ti)

  Extract representation label from tinfo key (as returned by return_type_tinfo()).
- static ex GiNaC::trace_string (exvector::const_iterator ix, size_t num)

  Take trace of a string of an even number of Dirac gammas given a vector of indices.
- ex GiNaC::dirac_trace (const ex &e, const std::set< unsigned char > &rls, const ex &trONE=4)

  Calculate dirac traces over the specified set of representation labels.
- ex GiNaC::dirac_trace (const ex &e, const lst &rll, const ex &trONE=4)

  Calculate dirac traces over the specified list of representation labels.
- ex GiNaC::dirac_trace (const ex &e, unsigned char rl=0, const ex &trONE=4)

  Calculate the trace of an expression containing gamma objects with a specified representation label.
- ex GiNaC::canonicalize_clifford (const ex &e)

  Bring all products of clifford objects in an expression into a canonical order.
- ex GiNaC::clifford_star_bar (const ex &e, bool do_bar, unsigned options)

  An auxillary function performing clifford_star() and clifford_bar().
- ex GiNaC::clifford_prime (const ex &e)

  Automorphism of the Clifford algebra, simply changes signs of all clifford units.
- ex GiNaC::remove_dirac_ONE (const ex &e, unsigned char rl=0, unsigned options=0)

  Replaces dirac_ONE's (with a representation_label no less than rl) in e with 1.
- int GiNaC::clifford_max_label (const ex &e, bool ignore_ONE=false)

  Returns the maximal representation label of a clifford object if e contains at least one, otherwise returns -1.
- ex GiNaC::clifford_norm (const ex &e)

Calculation of the norm in the Clifford algebra.

- ex GiNaC::clifford_inverse (const ex &e)

  Calculation of the inverse in the Clifford algebra.
- ex GiNaC::lst_to_clifford (const ex &v, const ex &mu, const ex &metr, unsigned char rl=0)

  List or vector conversion into the Clifford vector.
- ex GiNaC::lst_to_clifford (const ex &v, const ex &e)

  List or vector conversion into the Clifford vector.
- static ex GiNaC::get_clifford_comp (const ex &e, const ex &c, bool root=true)

  Auxiliary structure to define a function for striping one Clifford unit from vectors.
- lst GiNaC::clifford_to_lst (const ex &e, const ex &c, bool algebraic=true)

  An inverse function to lst_to_clifford().
- ex GiNaC::clifford_moebius_map (const ex &a, const ex &b, const ex &c, const ex &d, const ex &v, const ex &G, unsigned char rl=0)

  Calculations of Moebius transformations (conformal map) defined by a 2x2 Clifford matrix (a b\c d) in linear spaces with arbitrary signature.
- ex GiNaC::clifford_moebius_map (const ex &M, const ex &v, const ex &G, unsigned char rl=0)

  The second form of Moebius transformations defined by a 2x2 Clifford matrix M This function takes the transformation matrix M as a single entity.

Variables

- GiNaC::tensor

9.9.1 Detailed Description

Implementation of GiNaC's clifford algebra (Dirac gamma) objects.

9.10 clifford.h File Reference

Interface to GiNaC's clifford algebra (Dirac gamma) objects.

```
#include "indexed.h"
#include "tensor.h"
#include "symbol.h"
#include "idx.h"
#include <set>
```

Classes

- class GiNaC::clifford

  This class holds an object representing an element of the Clifford algebra (the Dirac gamma matrices).
- class GiNaC::diracone

  This class represents the Clifford algebra unity element.
- class GiNaC::cliffordunit

  This class represents the Clifford algebra generators (units).
- class GiNaC::diracgamma

  This class represents the Dirac gamma Lorentz vector.
- class GiNaC::diracgamma5

  This class represents the Dirac gamma5 object which anticommutates with all other gammas.
- class GiNaC::diracgammaL

  This class represents the Dirac gammaL object which behaves like 1/2 (1-gamma5).
- class GiNaC::diracgammaR

  This class represents the Dirac gammaL object which behaves like 1/2 (1+gamma5).

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (clifford)
- GiNaC::GINAC_DECLARE_UNARCHIVER (diracone)
- GiNaC::GINAC_DECLARE_UNARCHIVER (cliffordunit)
- GiNaC::GINAC_DECLARE_UNARCHIVER (diracgamma)
- GiNaC::GINAC_DECLARE_UNARCHIVER (diracgamma5)
- GiNaC::GINAC_DECLARE_UNARCHIVER (diracgammaL)
- GiNaC::GINAC_DECLARE_UNARCHIVER (diracgammaR)
- bool GiNaC::is_clifford_tinfo (const return_type_t &ti)

  Check whether a given return_type_t object (as returned by return_type_tinfo() is that of a clifford object (with an arbitrary representation label).
- ex GiNaC::dirac_ONE (unsigned char rl=0)

  Create a Clifford unity object.
- ex GiNaC::clifford_unit (const ex &mu, const ex &metr, unsigned char rl=0)

  Create a Clifford unit object.
- ex GiNaC::dirac_gamma (const ex &mu, unsigned char rl=0)

  Create a Dirac gamma object.
- ex GiNaC::dirac_gamma5 (unsigned char rl=0)

  Create a Dirac gamma5 object.
- ex GiNaC::dirac_gammaL (unsigned char rl=0)

  Create a Dirac gammaL object.
- ex GiNaC::dirac_gammaR (unsigned char rl=0)

  Create a Dirac gammaR object.
- ex GiNaC::dirac_slash (const ex &e, const ex &dim, unsigned char rl=0)

  *Create a term of the form e_mu * gamma~mu with a unique index mu.*
- ex GiNaC::dirac_trace (const ex &e, const std::set< unsigned char > &rls, const ex &trONE=4)

  Calculate dirac traces over the specified set of representation labels.
- ex GiNaC::dirac_trace (const ex &e, const lst &rll, const ex &trONE=4)

  Calculate dirac traces over the specified list of representation labels.
- ex GiNaC::dirac_trace (const ex &e, unsigned char rl=0, const ex &trONE=4)

  Calculate the trace of an expression containing gamma objects with a specified representation label.
- ex GiNaC::canonicalize_clifford (const ex &e)

  Bring all products of clifford objects in an expression into a canonical order.
- ex GiNaC::clifford_prime (const ex &e)

  Automorphism of the Clifford algebra, simply changes signs of all clifford units.
- ex GiNaC::clifford_star_bar (const ex &e, bool do_bar, unsigned options)

  An auxillary function performing clifford_star() and clifford_bar().
- ex GiNaC::clifford_bar (const ex &e)

  Main anti-automorphism of the Clifford algebra: makes reversion and changes signs of all clifford units.
- ex GiNaC::clifford_star (const ex &e)

  Reversion of the Clifford algebra, reverse the order of all clifford units in ncmul.
- ex GiNaC::remove_dirac_ONE (const ex &e, unsigned char rl=0, unsigned options=0)

  Replaces dirac_ONE's (with a representation_label no less than rl) in e with 1.
- int GiNaC::clifford_max_label (const ex &e, bool ignore_ONE=false)

  Returns the maximal representation label of a clifford object if e contains at least one, otherwise returns -1.
- ex GiNaC::clifford_norm (const ex &e)

Calculation of the norm in the Clifford algebra.

- ex GiNaC::clifford_inverse (const ex &e)

  Calculation of the inverse in the Clifford algebra.

- ex GiNaC::lst_to_clifford (const ex &v, const ex &mu, const ex &metr, unsigned char rl=0)

  List or vector conversion into the Clifford vector.

- ex GiNaC::lst_to_clifford (const ex &v, const ex &e)

  List or vector conversion into the Clifford vector.

- lst GiNaC::clifford_to_lst (const ex &e, const ex &c, bool algebraic=true)

  An inverse function to lst_to_clifford().

- ex GiNaC::clifford_moebius_map (const ex &a, const ex &b, const ex &c, const ex &d, const ex &v, const ex &G, unsigned char rl=0)

  Calculations of Moebius transformations (conformal map) defined by a 2x2 Clifford matrix (a b\c d) in linear spaces with arbitrary signature.

- ex GiNaC::clifford_moebius_map (const ex &M, const ex &v, const ex &G, unsigned char rl=0)

  The second form of Moebius transformations defined by a 2x2 Clifford matrix M This function takes the transformation matrix M as a single entity.

9.10.1 Detailed Description

Interface to GiNaC's clifford algebra (Dirac gamma) objects.

9.11 color.cpp File Reference

Implementation of GiNaC's color (SU(3) Lie algebra) objects.

```
#include "color.h"
#include "idx.h"
#include "ncmul.h"
#include "symmetry.h"
#include "operators.h"
#include "numeric.h"
#include "mul.h"
#include "power.h"
#include "symbol.h"
#include "archive.h"
#include "utils.h"
#include <stdexcept>
```

Namespaces

- namespace GiNaC

Macros

- #define TEST_PERMUTATION(A, B, C, P)
- #define CMPINDICES(A, B, C)

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (su3one, tensor, print_func< print_dflt >(&su3one::do_print). print_func< print_latex >(&su3one::do_print_latex)) GINAC_IMPLEMENT_↩ REGISTERED_CLASS_OPT(su3t
- GiNaC::print_func< print_dflt > (&su3t::do_print). print_func< print_latex >(&su3t
- GiNaC::GINAC_BIND_UNARCHIVER (color)
- GiNaC::GINAC_BIND_UNARCHIVER (su3one)
- GiNaC::GINAC_BIND_UNARCHIVER (su3t)
- GiNaC::GINAC_BIND_UNARCHIVER (su3f)
- GiNaC::GINAC_BIND_UNARCHIVER (su3d)
- static ex GiNaC::permute_free_index_to_front (const exvector &iv3, const exvector &iv2, int &sig)

  Given a vector iv3 of three indices and a vector iv2 of two indices that is a subset of iv3, return the (free) index that is in iv3 but not in iv2 and the sign introduced by permuting that index to the front.
- ex GiNaC::color_ONE (unsigned char rl=0)

  Create the su(3) unity element.
- ex GiNaC::color_T (const ex &a, unsigned char rl=0)

  Create an su(3) generator.
- ex GiNaC::color_f (const ex &a, const ex &b, const ex &c)

  Create an su(3) antisymmetric structure constant.
- ex GiNaC::color_d (const ex &a, const ex &b, const ex &c)

  Create an su(3) symmetric structure constant.
- ex GiNaC::color_h (const ex &a, const ex &b, const ex &c)

  This returns the linear combination d.a.b.c+I∗f.a.b.c.
- static bool GiNaC::is_color_tinfo (const return_type_t &ti)

  Check whether a given tinfo key (as returned by return_type_tinfo() is that of a color object (with an arbitrary representation label).
- static unsigned char GiNaC::get_representation_label (const return_type_t &ti)

  Extract representation label from tinfo key (as returned by return_type_tinfo()).
- ex GiNaC::color_trace (const ex &e, const std::set< unsigned char > &rls)

  Calculate color traces over the specified set of representation labels.
- ex GiNaC::color_trace (const ex &e, const lst &rll)

  Calculate color traces over the specified list of representation labels.
- ex GiNaC::color_trace (const ex &e, unsigned char rl=0)

  Calculate the trace of an expression containing color objects with a specified representation label.

9.11.1 Detailed Description

Implementation of GiNaC's color (SU(3) Lie algebra) objects.

9.11.2 Macro Definition Documentation

9.11.2.1 TEST_PERMUTATION

```
#define TEST_PERMUTATION(
               A,
               B,
               C,
               P)
```

Value:

```
if (iv3[B].is_equal(iv2[0]) && iv3[C].is_equal(iv2[1])) { \
    sig = P; \
    return iv3[A]; \
}
```

Referenced by GiNaC::permute_free_index_to_front().

9.11.2.2 CMPINDICES

```
#define CMPINDICES(
              A,
              B,
              C)
```

Value:

```
((v[0] == (A)) && (v[1] == (B)) && (v[2] == (C)))
```

Referenced by GiNaC::su3d::eval_indexed(), and GiNaC::su3f::eval_indexed().

9.12 color.h File Reference

Interface to GiNaC's color (SU(3) Lie algebra) objects.

```
#include "indexed.h"
#include "tensor.h"
#include <set>
```

Classes

- class GiNaC::color

  This class holds a generator T_a or the unity element of the Lie algebra of SU(3), as used for calculations in quantum chromodynamics.
- class GiNaC::su3one

  This class represents the su(3) unity element.
- class GiNaC::su3t

  This class represents an su(3) generator.
- class GiNaC::su3f

  This class represents the tensor of antisymmetric su(3) structure constants.
- class GiNaC::su3d

  This class represents the tensor of symmetric su(3) structure constants.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (color)
- GiNaC::GINAC_DECLARE_UNARCHIVER (su3one)
- GiNaC::GINAC_DECLARE_UNARCHIVER (su3t)
- GiNaC::GINAC_DECLARE_UNARCHIVER (su3f)
- GiNaC::GINAC_DECLARE_UNARCHIVER (su3d)
- ex GiNaC::color_ONE (unsigned char rl=0)

  Create the su(3) unity element.
- ex GiNaC::color_T (const ex &a, unsigned char rl=0)

  Create an su(3) generator.
- ex GiNaC::color_f (const ex &a, const ex &b, const ex &c)

  Create an su(3) antisymmetric structure constant.
- ex GiNaC::color_d (const ex &a, const ex &b, const ex &c)

  Create an su(3) symmetric structure constant.
- ex GiNaC::color_h (const ex &a, const ex &b, const ex &c)

  This returns the linear combination d.a.b.c+I∗f.a.b.c.
- ex GiNaC::color_trace (const ex &e, const std::set< unsigned char > &rls)

  Calculate color traces over the specified set of representation labels.
- ex GiNaC::color_trace (const ex &e, const lst &rll)

  Calculate color traces over the specified list of representation labels.
- ex GiNaC::color_trace (const ex &e, unsigned char rl=0)

  Calculate the trace of an expression containing color objects with a specified representation label.

9.12.1 Detailed Description

Interface to GiNaC's color (SU(3) Lie algebra) objects.

9.13 compiler.h File Reference

Definition of optimizing macros.

Macros

- #define unlikely(cond)
- #define likely(cond)
- #define attribute_pure

9.13.1 Detailed Description

Definition of optimizing macros.

9.13.2 Macro Definition Documentation

9.13.2.1 unlikely

```
#define unlikely(
              cond)
```

Value:

```
(cond)
```

Referenced by GiNaC::add::eval(), GiNaC::mul::eval(), and GiNaC::expairseq::evalchildren().

9.13.2.2 likely

```
#define likely(
              cond)
```

Value:

```
(cond)
```

Referenced by GiNaC::add::combine_ex_with_coeff_to_pair(), GiNaC::mul::eval(), and GiNaC::power::eval().

9.13.2.3 attribute_pure

```
#define attribute_pure
```

9.14 constant.cpp File Reference

Implementation of GiNaC's constant types and some special constants.

```
#include "constant.h"
#include "numeric.h"
#include "ex.h"
#include "archive.h"
#include "utils.h"
#include "inifcns.h"
#include <stdexcept>
#include <string>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (constant, basic, print_func< print_context >(&constant::do_print). print_func< print_latex >(&constant::do_print_latex). print_func< print_tree >(&constant::do_print_tree). print_func< print_python_repr >(&constant::do_print_python_repr)) const ant
- GiNaC::GINAC_BIND_UNARCHIVER (constant)

Variables

- const constant GiNaC::Pi ("Pi", PiEvalf, "\\pi", domain::positive)

  Pi.
- const constant GiNaC::Euler ("Euler", EulerEvalf, "\\gamma_E", domain::positive)

  Euler's constant.
- const constant GiNaC::Catalan ("Catalan", CatalanEvalf, "G", domain::positive)

  Catalan's constant.

9.14.1 Detailed Description

Implementation of GiNaC's constant types and some special constants.

9.15 constant.h File Reference

Interface to GiNaC's constant types and some special constants.

```
#include "basic.h"
#include "ex.h"
#include "archive.h"
#include <string>
```

Classes

- class GiNaC::constant

  This class holds constants, symbols with specific numerical value.

Namespaces

- namespace GiNaC

Typedefs

- typedef ex(∗) GiNaC::evalffunctype()

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (constant)

9.15.1 Detailed Description

Interface to GiNaC's constant types and some special constants.

9.16 container.h File Reference

Wrapper template for making GiNaC classes out of STL containers.

```
#include "ex.h"
#include "print.h"
#include "archive.h"
#include "assertion.h"
#include "compiler.h"
#include <algorithm>
#include <iterator>
#include <list>
#include <memory>
#include <stdexcept>
#include <vector>
```

Classes

- class GiNaC::container_storage< C >

  Helper template for encapsulating the reserve() mechanics of STL containers.
- class GiNaC::container< C >

  Wrapper template for making GiNaC classes out of STL containers.

Namespaces

- namespace GiNaC

9.16.1 Detailed Description

Wrapper template for making GiNaC classes out of STL containers.

9.17 crc32.h File Reference

CRC32 hash function.

Namespaces

- namespace GiNaC

Functions

- static unsigned GiNaC::crc32 (const char *c, const unsigned len, const unsigned crcinit)

Variables

- static unsigned const GiNaC::crctab [256]

9.17.1 Detailed Description

CRC32 hash function.

Shamelessly stolen from GNU coreutils (cksum.c).

9.18 ex.cpp File Reference

Implementation of GiNaC's light-weight expression handles.

```
#include "ex.h"
#include "add.h"
#include "mul.h"
#include "ncmul.h"
#include "numeric.h"
#include "matrix.h"
#include "power.h"
#include "lst.h"
#include "relational.h"
#include "utils.h"
#include <stdexcept>
```

Namespaces

- namespace GiNaC

9.18.1 Detailed Description

Implementation of GiNaC's light-weight expression handles.

9.19 ex.h File Reference

Interface to GiNaC's light-weight expression handles.

```
#include "basic.h"
#include "ptr.h"
#include "compiler.h"
#include <functional>
#include <iosfwd>
#include <iterator>
#include <memory>
#include <stack>
```

Classes

- class GiNaC::library_init

  Helper class to initialize the library.
- class GiNaC::ex

  Lightweight wrapper for GiNaC's symbolic objects.
- class GiNaC::const_iterator
- struct GiNaC::internal::_iter_rep
- class GiNaC::const_preorder_iterator
- class GiNaC::const_postorder_iterator
- struct GiNaC::ex_is_less
- struct GiNaC::ex_is_equal
- struct GiNaC::op0_is_equal
- struct GiNaC::ex_swap
- class GiNaC::pointer_to_map_function
- class GiNaC::pointer_to_map_function_1arg< T1 >
- class GiNaC::pointer_to_map_function_2args< T1, T2 >
- class GiNaC::pointer_to_map_function_3args< T1, T2, T3 >
- class GiNaC::pointer_to_member_to_map_function< C >
- class GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >
- class GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >
- class GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >
- struct std::hash< GiNaC::ex >

  Specialization of std::hash() for ex objects.
- struct std::equal_to< GiNaC::ex >

  Specialization of std::equal_to() for ex objects.

Namespaces

- namespace GiNaC
- namespace GiNaC::internal
- namespace std

Functions

- bool GiNaC::are_ex_trivially_equal (const ex &e1, const ex &e2)

  Compare two objects of class quickly without doing a deep tree traversal.
- std::ostream & GiNaC::operator<< (std::ostream &os, const exvector &e)
- std::ostream & GiNaC::operator<< (std::ostream &os, const exset &e)
- std::ostream & GiNaC::operator<< (std::ostream &os, const exmap &e)
- size_t GiNaC::nops (const ex &thisex)
- ex GiNaC::expand (const ex &thisex, unsigned options=0)
- ex GiNaC::conjugate (const ex &thisex)
- ex GiNaC::real_part (const ex &thisex)
- ex GiNaC::imag_part (const ex &thisex)
- bool GiNaC::has (const ex &thisex, const ex &pattern, unsigned options=0)
- bool GiNaC::find (const ex &thisex, const ex &pattern, exset &found)
- bool GiNaC::is_polynomial (const ex &thisex, const ex &vars)
- int GiNaC::degree (const ex &thisex, const ex &s)
- int GiNaC::ldegree (const ex &thisex, const ex &s)
- ex GiNaC::coeff (const ex &thisex, const ex &s, int n=1)
- ex GiNaC::numer (const ex &thisex)

- ex GiNaC::denom (const ex &thisex)
- ex GiNaC::numer_denom (const ex &thisex)
- ex GiNaC::normal (const ex &thisex)
- ex GiNaC::to_rational (const ex &thisex, exmap &repl)
- ex GiNaC::to_polynomial (const ex &thisex, exmap &repl)
- ex GiNaC::collect (const ex &thisex, const ex &s, bool distributed=false)
- ex GiNaC::eval (const ex &thisex)
- ex GiNaC::evalf (const ex &thisex)
- ex GiNaC::evalm (const ex &thisex)
- ex GiNaC::eval_integ (const ex &thisex)
- ex GiNaC::diff (const ex &thisex, const symbol &s, unsigned nth=1)
- ex GiNaC::series (const ex &thisex, const ex &r, int order, unsigned options=0)
- bool GiNaC::match (const ex &thisex, const ex &pattern, exmap &repl_lst)
- ex GiNaC::simplify_indexed (const ex &thisex, unsigned options=0)
- ex GiNaC::simplify_indexed (const ex &thisex, const scalar_products &sp, unsigned options=0)
- ex GiNaC::symmetrize (const ex &thisex)
- ex GiNaC::symmetrize (const ex &thisex, const lst &l)
- ex GiNaC::antisymmetrize (const ex &thisex)
- ex GiNaC::antisymmetrize (const ex &thisex, const lst &l)
- ex GiNaC::symmetrize_cyclic (const ex &thisex)
- ex GiNaC::symmetrize_cyclic (const ex &thisex, const lst &l)
- ex GiNaC::op (const ex &thisex, size_t i)
- ex GiNaC::lhs (const ex &thisex)
- ex GiNaC::rhs (const ex &thisex)
- bool GiNaC::is_zero (const ex &thisex)
- void GiNaC::swap (ex &e1, ex &e2)
- ex GiNaC::subs (const ex &thisex, const exmap &m, unsigned options=0)
- ex GiNaC::subs (const ex &thisex, const lst &ls, const lst &lr, unsigned options=0)
- ex GiNaC::subs (const ex &thisex, const ex &e, unsigned options=0)
- template<class T>
  bool GiNaC::is_a (const ex &obj)

  Check if ex is a handle to a T, including base classes.
- template<class T>
  bool GiNaC::is_exactly_a (const ex &obj)

  Check if ex is a handle to a T, not including base classes.
- template<class T>
  attribute_pure const T & GiNaC::ex_to (const ex &e)

  Return a reference to the basic-derived class T object embedded in an expression.
- template<> void std::swap (GiNaC::ex &a, GiNaC::ex &b)

  Specialization of std::swap() for ex objects.

Variables

- static library_init GiNaC::library_initializer

  For construction of flyweights, etc.
- const basic * GiNaC::_num0_bp

9.19.1 Detailed Description

Interface to GiNaC's light-weight expression handles.

9.20 excompiler.cpp File Reference

Functions to facilitate the conversion of a ex to a function pointer suited for fast numerical integration.

```
#include "excompiler.h"
#include "ex.h"
#include "lst.h"
#include "operators.h"
#include "relational.h"
#include "symbol.h"
#include <cstdlib>
#include <fstream>
#include <ios>
#include <sstream>
#include <stdexcept>
#include <string>
#include <vector>
```

Namespaces

- namespace GiNaC

Functions

- void GiNaC::compile_ex (const ex &expr, const symbol &sym, FUNCP_1P &fp, const std::string filename="")

  Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.
- void GiNaC::compile_ex (const ex &expr, const symbol &sym1, const symbol &sym2, FUNCP_2P &fp, const std::string filename="")

  Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.
- void GiNaC::compile_ex (const lst &exprs, const lst &syms, FUNCP_CUBA &fp, const std::string filename="")

  Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.
- void GiNaC::link_ex (const std::string filename, FUNCP_1P &fp)

  Opens an existing so-file and returns a function pointer of type FUNCP_1P to the contained function.
- void GiNaC::link_ex (const std::string filename, FUNCP_CUBA &fp)

  Opens an existing so-file and returns a function pointer of type FUNCP_CUBA to the contained function.
- void GiNaC::unlink_ex (const std::string filename)

  Closes all linked .so files that have the supplied filename.

9.20.1 Detailed Description

Functions to facilitate the conversion of a ex to a function pointer suited for fast numerical integration.

9.21 excompiler.h File Reference

Functions to facilitate the conversion of a ex to a function pointer suited for fast numerical integration.

```
#include "lst.h"
#include <string>
```

Namespaces

- namespace GiNaC

Typedefs

- typedef double(∗) GiNaC::FUNCP_1P(double)

  Function pointer with one function parameter.
- typedef double(∗) GiNaC::FUNCP_2P(double, double)

  Function pointer with two function parameters.
- typedef void(∗) GiNaC::FUNCP_CUBA(const int ∗, const double[ ], const int ∗, double[ ])

  Function pointer for use with the CUBA library (http://www.feynarts.de/cuba).

Functions

- void GiNaC::compile_ex (const ex &expr, const symbol &sym, FUNCP_1P &fp, const std::string filename="")

  Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.
- void GiNaC::compile_ex (const ex &expr, const symbol &sym1, const symbol &sym2, FUNCP_2P &fp, const std::string filename="")

  Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.
- void GiNaC::compile_ex (const lst &exprs, const lst &syms, FUNCP_CUBA &fp, const std::string filename="")

  Takes an expression and produces a function pointer to the compiled and linked C code equivalent in double precision.
- void GiNaC::link_ex (const std::string filename, FUNCP_1P &fp)

  Opens an existing so-file and returns a function pointer of type FUNCP_1P to the contained function.
- void GiNaC::link_ex (const std::string filename, FUNCP_CUBA &fp)

  Opens an existing so-file and returns a function pointer of type FUNCP_CUBA to the contained function.
- void GiNaC::unlink_ex (const std::string filename)

  Closes all linked .so files that have the supplied filename.

9.21.1 Detailed Description

Functions to facilitate the conversion of a ex to a function pointer suited for fast numerical integration.

9.22 expair.cpp File Reference

Implementation of expression pairs (building blocks of expairseq).

```
#include "expair.h"
#include "operators.h"
```

Namespaces

- namespace GiNaC

9.22.1 Detailed Description

Implementation of expression pairs (building blocks of expairseq).

9.23 expair.h File Reference

Definition of expression pairs (building blocks of expairseq).

```
#include "ex.h"
#include "numeric.h"
#include "print.h"
```

Classes

- class GiNaC::expair
  A pair of expressions.
- struct GiNaC::expair_is_less
  Function object for insertion into third argument of STL's sort() etc.
- struct GiNaC::expair_rest_is_less
  Function object not caring about the numerical coefficients for insertion into third argument of STL's sort().
- struct GiNaC::expair_swap

Namespaces

- namespace GiNaC

Functions

- void GiNaC::swap (expair &e1, expair &e2)

9.23.1 Detailed Description

Definition of expression pairs (building blocks of expairseq).

9.24 expairseq.cpp File Reference

Implementation of sequences of expression pairs.

```
#include "expairseq.h"
#include "lst.h"
#include "add.h"
#include "mul.h"
#include "power.h"
#include "relational.h"
#include "wildcard.h"
#include "archive.h"
#include "operators.h"
#include "utils.h"
#include "hash_seed.h"
#include "indexed.h"
#include "compiler.h"
#include <algorithm>
#include <iostream>
#include <iterator>
#include <memory>
#include <stdexcept>
#include <string>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (expairseq, basic, print_func< print_context >(&expairseq::do_print). print_func< print_tree >(&expairseq::do_print_tree)) class epp_is_less
- epvector * GiNaC::conjugateepvector (const epvector &)

  Complex conjugate every element of an epvector.

9.24.1 Detailed Description

Implementation of sequences of expression pairs.

9.25 expairseq.h File Reference

Interface to sequences of expression pairs.

```
#include "expair.h"
#include "indexed.h"
#include <vector>
```

Classes

- class GiNaC::expairseq

  A sequence of class expair.
- class GiNaC::make_flat_inserter

  Class to handle the renaming of dummy indices.

Namespaces

- namespace GiNaC

Typedefs

- typedef std::vector< expair > GiNaC::epvector

  expair-vector
- typedef epvector::iterator GiNaC::epp

  expair-vector pointer

Functions

- epvector * GiNaC::conjugateepvector (const epvector &)

  Complex conjugate every element of an epvector.

9.25.1 Detailed Description

Interface to sequences of expression pairs.

9.26 exprseq.cpp File Reference

Implementation of GiNaC's exprseq.

```
#include "exprseq.h"
```

Namespaces

- namespace GiNaC

9.26.1 Detailed Description

Implementation of GiNaC's exprseq.

9.27 exprseq.h File Reference

Definition of GiNaC's exprseq.

```
#include "container.h"
#include <vector>
```

Namespaces

- namespace GiNaC

Typedefs

- typedef container< std::vector > GiNaC::exprseq

9.27.1 Detailed Description

Definition of GiNaC's exprseq.

9.28 factor.cpp File Reference

Polynomial factorization (implementation).

```
#include "factor.h"
#include "ex.h"
#include "numeric.h"
#include "operators.h"
#include "inifcns.h"
#include "symbol.h"
#include "relational.h"
#include "power.h"
#include "mul.h"
#include "normal.h"
#include "add.h"
#include <type_traits>
#include <algorithm>
#include <limits>
#include <list>
#include <vector>
#include <stack>
#include <cln/cln.h>
```

Namespaces

- namespace GiNaC

Macros

- #define DCOUT(str)
- #define DCOUTVAR(var)
- #define DCOUT2(str, var)
- #define USE_SAME_DEGREE_FACTOR

Functions

- ex GiNaC::factor (const ex &poly, unsigned options)

  Interface function to the outside world.

9.28.1 Detailed Description

Polynomial factorization (implementation).

The interface function factor() at the end of this file is defined in the GiNaC namespace. All other utility functions and classes are defined in an additional anonymous namespace.

Factorization starts by doing a square free factorization and making the coefficients integer. Then, depending on the number of free variables it proceeds either in dedicated univariate or multivariate factorization code.

Univariate factorization does a modular factorization via Berlekamp's algorithm and distinct degree factorization. Hensel lifting is used at the end.

Multivariate factorization uses the univariate factorization (applying a evaluation homomorphism first) and Hensel lifting raises the answer to the multivariate domain. The Hensel lifting code is completely distinct from the code used by the univariate factorization.

Algorithms used can be found in [Wan] An Improved Multivariate Polynomial Factoring Algorithm, P.S.Wang, Mathematics of Computation, Vol. 32, No. 144 (1978) 1215–1231. [GCL] Algorithms for Computer Algebra, K.O.Geddes, S.R.Czapor, G.Labahn, Springer Verlag, 1992. [Mig] Some Useful Bounds, M.Mignotte, In "Computer Algebra, Symbolic and Algebraic Computation" (B.Buchberger et al., eds.), pp. 259-263, Springer-Verlag, New York, 1982.

9.28.2 Macro Definition Documentation

9.28.2.1 DCOUT

```
#define DCOUT(
              str)
```

9.28.2.2 DCOUTVAR

```
#define DCOUTVAR(
              var)
```

9.28.2.3 DCOUT2

```
#define DCOUT2(
              str,
              var)
```

9.28.2.4 USE_SAME_DEGREE_FACTOR

```
#define USE_SAME_DEGREE_FACTOR
```

9.29 factor.h File Reference

Polynomial factorization.

Namespaces

- namespace GiNaC

Functions

- ex GiNaC::factor (const ex &poly, unsigned options)

  Interface function to the outside world.

9.29.1 Detailed Description

Polynomial factorization.

9.30 fail.cpp File Reference

Implementation of class signaling failure of operation.

```
#include "fail.h"
#include "archive.h"
#include "utils.h"
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (fail, basic, print_func< print_context >(&fail::do_print). print_func< print_tree >(&fail::do_print_tree)) GINAC_BIND_UNARCHIVER(fail)

9.30.1 Detailed Description

Implementation of class signaling failure of operation.

Considered somewhat obsolete (most of this can be replaced by exceptions).

9.31 fail.h File Reference

Interface to class signaling failure of operation.

```
#include "basic.h"
#include "archive.h"
```

Classes

- class GiNaC::fail

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (fail)

9.31.1 Detailed Description

Interface to class signaling failure of operation.

Considered obsolete somewhat obsolete (most of this can be replaced by exceptions).

9.32 fderivative.cpp File Reference

Implementation of abstract derivatives of functions.

```
#include "fderivative.h"
#include "operators.h"
#include "archive.h"
#include "utils.h"
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (fderivative, function, print_func< print_context >(&fderivative::do_print). print_func< print_latex >(&fderivative::do_print_latex). print_func< print_csrc >(&fderivative::do_print_csrc). print_func< print_tree >(&fderivative::do_print_tree)) fderivative
- GiNaC::GINAC_BIND_UNARCHIVER (fderivative)

9.32.1 Detailed Description

Implementation of abstract derivatives of functions.

9.33 fderivative.h File Reference

Interface to abstract derivatives of functions.

```
#include "function.h"
#include <set>
```

Classes

- class GiNaC::fderivative

  This class represents the (abstract) derivative of a symbolic function.

Namespaces

- namespace GiNaC

Typedefs

- typedef std::multiset< unsigned > GiNaC::paramset

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (fderivative)

9.33.1 Detailed Description

Interface to abstract derivatives of functions.

9.34 flags.h File Reference

Collection of all flags used through the GiNaC framework.

Classes

- class GiNaC::expand_options

  Flags to control the behavior of expand().
- class GiNaC::has_options

  Flags to control the behavior of has().
- class GiNaC::subs_options

  Flags to control the behavior of subs().
- class GiNaC::domain

  Domain of an object.
- class GiNaC::series_options

  Flags to control series expansion.
- class GiNaC::determinant_algo

  Switch to control algorithm for determinant computation.
- class GiNaC::solve_algo

  Switch to control algorithm for linear system solving.
- class GiNaC::status_flags

  Flags to store information about the state of an object.
- class GiNaC::info_flags

  Possible attributes an object can have.
- class GiNaC::return_types
- class GiNaC::remember_strategies

  Strategies how to clean up the function remember cache.
- class GiNaC::factor_options

  Flags to control the polynomial factorization.

Namespaces

- namespace GiNaC

9.34.1 Detailed Description

Collection of all flags used through the GiNaC framework.

9.35 function.cpp File Reference

Implementation of class of symbolic functions.

```
#include "function.h"
#include "operators.h"
#include "fderivative.h"
#include "ex.h"
#include "lst.h"
#include "symmetry.h"
#include "print.h"
#include "power.h"
#include "archive.h"
#include "inifcns.h"
#include "utils.h"
```

```
#include "hash_seed.h"
#include "remember.h"
#include <iostream>
#include <limits>
#include <list>
#include <stdexcept>
#include <string>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_BIND_UNARCHIVER (function)

9.35.1 Detailed Description

Implementation of class of symbolic functions.

9.36 function.h File Reference

Interface to class of symbolic functions.

```
#include "exprseq.h"
#include <string>
#include <vector>
```

Classes

- class GiNaC::function_options
- class GiNaC::do_taylor

  Exception class thrown by classes which provide their own series expansion to signal that ordinary Taylor expansion is safe.
- class GiNaC::function

  The class function is used to implement builtin functions like sin, cos... and user defined functions.

Namespaces

- namespace GiNaC

Macros

- #define DECLARE_FUNCTION_1P(NAME)
- #define DECLARE_FUNCTION_2P(NAME)
- #define DECLARE_FUNCTION_3P(NAME)
- #define DECLARE_FUNCTION_4P(NAME)
- #define DECLARE_FUNCTION_5P(NAME)
- #define DECLARE_FUNCTION_6P(NAME)
- #define DECLARE_FUNCTION_7P(NAME)
- #define DECLARE_FUNCTION_8P(NAME)
- #define DECLARE_FUNCTION_9P(NAME)
- #define DECLARE_FUNCTION_10P(NAME)
- #define DECLARE_FUNCTION_11P(NAME)
- #define DECLARE_FUNCTION_12P(NAME)
- #define DECLARE_FUNCTION_13P(NAME)
- #define DECLARE_FUNCTION_14P(NAME)
- #define REGISTER_FUNCTION(NAME, OPT)
- #define is_ex_the_function(OBJ, FUNCNAME)

Typedefs

- typedef ex(∗) GiNaC::eval_funcp()
- typedef ex(∗) GiNaC::evalf_funcp()
- typedef ex(∗) GiNaC::conjugate_funcp()
- typedef ex(∗) GiNaC::real_part_funcp()
- typedef ex(∗) GiNaC::imag_part_funcp()
- typedef ex(∗) GiNaC::expand_funcp()
- typedef ex(∗) GiNaC::derivative_funcp()
- typedef ex(∗) GiNaC::expl_derivative_funcp()
- typedef ex(∗) GiNaC::power_funcp()
- typedef ex(∗) GiNaC::series_funcp()
- typedef void(∗) GiNaC::print_funcp()
- typedef bool(∗) GiNaC::info_funcp()
- typedef ex(∗) GiNaC::eval_funcp_1(const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_1(const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_1(const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_1(const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_1(const ex &)
- typedef ex(∗) GiNaC::expand_funcp_1(const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_1(const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_1(const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_1(const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_1(const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_1(const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_1(const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_2(const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_2(const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_2(const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_2(const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_2(const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_2(const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_2(const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_2(const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_2(const ex &, const ex &, const ex &)

- typedef ex(∗) GiNaC::series_funcp_2(const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_2(const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_2(const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_3(const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_3(const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_3(const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_3(const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_3(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_3(const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_3(const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_3(const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_4(const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_4(const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_4(const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_4(const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_4(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_4(const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_4(const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_4(const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_5(const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)

- typedef ex(∗) GiNaC::derivative_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_6(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_7(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)

- typedef void(∗) GiNaC::print_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_8(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_9(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_10(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)

- typedef ex(∗) GiNaC::conjugate_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_11(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_12(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)

- typedef ex(∗) GiNaC::derivative_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_13(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::evalf_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::conjugate_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::real_part_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::imag_part_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::expand_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_14(const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, const ex &, unsigned)
- typedef ex(∗) GiNaC::eval_funcp_exvector(const exvector &)
- typedef ex(∗) GiNaC::evalf_funcp_exvector(const exvector &)
- typedef ex(∗) GiNaC::conjugate_funcp_exvector(const exvector &)
- typedef ex(∗) GiNaC::real_part_funcp_exvector(const exvector &)
- typedef ex(∗) GiNaC::imag_part_funcp_exvector(const exvector &)
- typedef ex(∗) GiNaC::expand_funcp_exvector(const exvector &, unsigned)
- typedef ex(∗) GiNaC::derivative_funcp_exvector(const exvector &, unsigned)
- typedef ex(∗) GiNaC::expl_derivative_funcp_exvector(const exvector &, const symbol &)
- typedef ex(∗) GiNaC::power_funcp_exvector(const exvector &, const ex &)
- typedef ex(∗) GiNaC::series_funcp_exvector(const exvector &, const relational &, int, unsigned)
- typedef void(∗) GiNaC::print_funcp_exvector(const exvector &, const print_context &)
- typedef bool(∗) GiNaC::info_funcp_exvector(const exvector &, unsigned)

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (function)
- template<typename T>
  bool GiNaC::is_the_function (const ex &x)

9.36.1 Detailed Description

Interface to class of symbolic functions.

9.36.2 Macro Definition Documentation

9.36.2.1 DECLARE_FUNCTION_1P

```
#define DECLARE_FUNCTION_1P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 1; \
template< typename T1 > const GiNaC::function NAME( const T1 & p1 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1) ); \
}
```

9.36.2.2 DECLARE_FUNCTION_2P

```
#define DECLARE_FUNCTION_2P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 2; \
template< typename T1, typename T2 > const GiNaC::function NAME( const T1 & p1, const T2 & p2 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2) ); \
}
```

9.36.2.3 DECLARE_FUNCTION_3P

```
#define DECLARE_FUNCTION_3P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 3; \
template< typename T1, typename T2, typename T3 > const GiNaC::function NAME( const T1 & p1, const T2 & p2,
      const T3 & p3 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3) ); \
}
```

9.36.2.4 DECLARE_FUNCTION_4P

```
#define DECLARE_FUNCTION_4P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 4; \
template< typename T1, typename T2, typename T3, typename T4 > const GiNaC::function NAME( const T1 & p1,
      const T2 & p2, const T3 & p3, const T4 & p4 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3), GiNaC::ex(p4)
      ); \
}
```

9.36.2.5 DECLARE_FUNCTION_5P

```
#define DECLARE_FUNCTION_5P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 5; \
template< typename T1, typename T2, typename T3, typename T4, typename T5 > const GiNaC::function NAME(
      const T1 & p1, const T2 & p2, const T3 & p3, const T4 & p4, const T5 & p5 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5) ); \
}
```

9.36.2.6 DECLARE_FUNCTION_6P

```
#define DECLARE_FUNCTION_6P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 6; \
template< typename T1, typename T2, typename T3, typename T4, typename T5, typename T6 > const
      GiNaC::function NAME( const T1 & p1, const T2 & p2, const T3 & p3, const T4 & p4, const T5 & p5, const
      T6 & p6 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5), GiNaC::ex(p6) ); \
}
```

9.36.2.7 DECLARE_FUNCTION_7P

```
#define DECLARE_FUNCTION_7P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 7; \
template< typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7 > const
      GiNaC::function NAME( const T1 & p1, const T2 & p2, const T3 & p3, const T4 & p4, const T5 & p5, const
      T6 & p6, const T7 & p7 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5), GiNaC::ex(p6), GiNaC::ex(p7) ); \
}
```

9.36.2.8 DECLARE_FUNCTION_8P

```
#define DECLARE_FUNCTION_8P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 8; \
template< typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7,
      typename T8 > const GiNaC::function NAME( const T1 & p1, const T2 & p2, const T3 & p3, const T4 & p4,
      const T5 & p5, const T6 & p6, const T7 & p7, const T8 & p8 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5), GiNaC::ex(p6), GiNaC::ex(p7), GiNaC::ex(p8) ); \
}
```

9.36.2.9 DECLARE_FUNCTION_9P

```
#define DECLARE_FUNCTION_9P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 9; \
template< typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7,
      typename T8, typename T9 > const GiNaC::function NAME( const T1 & p1, const T2 & p2, const T3 & p3,
      const T4 & p4, const T5 & p5, const T6 & p6, const T7 & p7, const T8 & p8, const T9 & p9 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5), GiNaC::ex(p6), GiNaC::ex(p7), GiNaC::ex(p8), GiNaC::ex(p9) ); \
}
```

9.36.2.10 DECLARE_FUNCTION_10P

```
#define DECLARE_FUNCTION_10P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 10; \
template< typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7,
      typename T8, typename T9, typename T10 > const GiNaC::function NAME( const T1 & p1, const T2 & p2,
      const T3 & p3, const T4 & p4, const T5 & p5, const T6 & p6, const T7 & p7, const T8 & p8, const T9 &
      p9, const T10 & p10 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5), GiNaC::ex(p6), GiNaC::ex(p7), GiNaC::ex(p8), GiNaC::ex(p9),
      GiNaC::ex(p10) ); \
}
```

9.36.2.11 DECLARE_FUNCTION_11P

```
#define DECLARE_FUNCTION_11P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 11; \
template< typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7,
      typename T8, typename T9, typename T10, typename T11 > const GiNaC::function NAME( const T1 & p1,
      const T2 & p2, const T3 & p3, const T4 & p4, const T5 & p5, const T6 & p6, const T7 & p7, const T8 &
      p8, const T9 & p9, const T10 & p10, const T11 & p11 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5), GiNaC::ex(p6), GiNaC::ex(p7), GiNaC::ex(p8), GiNaC::ex(p9),
      GiNaC::ex(p10), GiNaC::ex(p11) ); \
}
```

9.36.2.12 DECLARE_FUNCTION_12P

```
#define DECLARE_FUNCTION_12P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 12; \
template< typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7,
      typename T8, typename T9, typename T10, typename T11, typename T12 > const GiNaC::function NAME( const
      T1 & p1, const T2 & p2, const T3 & p3, const T4 & p4, const T5 & p5, const T6 & p6, const T7 & p7,
      const T8 & p8, const T9 & p9, const T10 & p10, const T11 & p11, const T12 & p12 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5), GiNaC::ex(p6), GiNaC::ex(p7), GiNaC::ex(p8), GiNaC::ex(p9),
      GiNaC::ex(p10), GiNaC::ex(p11), GiNaC::ex(p12) ); \
}
```

9.36.2.13 DECLARE_FUNCTION_13P

```
#define DECLARE_FUNCTION_13P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 13; \
template< typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7,
      typename T8, typename T9, typename T10, typename T11, typename T12, typename T13 > const
      GiNaC::function NAME( const T1 & p1, const T2 & p2, const T3 & p3, const T4 & p4, const T5 & p5, const
      T6 & p6, const T7 & p7, const T8 & p8, const T9 & p9, const T10 & p10, const T11 & p11, const T12 &
      p12, const T13 & p13 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5), GiNaC::ex(p6), GiNaC::ex(p7), GiNaC::ex(p8), GiNaC::ex(p9),
      GiNaC::ex(p10), GiNaC::ex(p11), GiNaC::ex(p12), GiNaC::ex(p13) ); \
}
```

9.36.2.14 DECLARE_FUNCTION_14P

```
#define DECLARE_FUNCTION_14P(
                NAME)
```

Value:

```
class NAME##_SERIAL { public: static unsigned serial; }; \
const unsigned NAME##_NPARAMS = 14; \
template< typename T1, typename T2, typename T3, typename T4, typename T5, typename T6, typename T7,
      typename T8, typename T9, typename T10, typename T11, typename T12, typename T13, typename T14 > const
      GiNaC::function NAME( const T1 & p1, const T2 & p2, const T3 & p3, const T4 & p4, const T5 & p5, const
      T6 & p6, const T7 & p7, const T8 & p8, const T9 & p9, const T10 & p10, const T11 & p11, const T12 &
      p12, const T13 & p13, const T14 & p14 ) { \
    return GiNaC::function(NAME##_SERIAL::serial, GiNaC::ex(p1), GiNaC::ex(p2), GiNaC::ex(p3),
      GiNaC::ex(p4), GiNaC::ex(p5), GiNaC::ex(p6), GiNaC::ex(p7), GiNaC::ex(p8), GiNaC::ex(p9),
      GiNaC::ex(p10), GiNaC::ex(p11), GiNaC::ex(p12), GiNaC::ex(p13), GiNaC::ex(p14) ); \
}
```

9.36.2.15 REGISTER_FUNCTION

```
#define REGISTER_FUNCTION(
                NAME,
                OPT)
```

Value:

```
unsigned NAME##_SERIAL::serial = \
    GiNaC::function::register_new(GiNaC::function_options(#NAME, NAME##_NPARAMS).OPT);
```

9.36.2.16 is_ex_the_function

```
#define is_ex_the_function(
              OBJ,
              FUNCNAME)
```

Value:

```
(GiNaC::is_the_function<FUNCNAME##_SERIAL>(OBJ))
```

Referenced by GiNaC::abs_eval(), GiNaC::cos_eval(), GiNaC::cosh_eval(), GiNaC::exp_eval(), GiNaC::is_order_function(), GiNaC::log_eval(), GiNaC::replace_with_symbol(), GiNaC::sin_eval(), GiNaC::sinh_eval(), GiNaC::tan_eval(), and GiNaC::tanh_eval().

9.37 ginac.h File Reference

This include file includes all other public GiNaC headers.

```
#include "version.h"
#include "basic.h"
#include "ex.h"
#include "normal.h"
#include "archive.h"
#include "print.h"
#include "constant.h"
#include "fail.h"
#include "integral.h"
#include "lst.h"
#include "matrix.h"
#include "numeric.h"
#include "power.h"
#include "relational.h"
#include "structure.h"
#include "symbol.h"
#include "pseries.h"
#include "wildcard.h"
#include "symmetry.h"
#include "expair.h"
#include "expairseq.h"
#include "add.h"
#include "mul.h"
#include "exprseq.h"
#include "function.h"
#include "ncmul.h"
#include "inifcns.h"
#include "fderivative.h"
#include "operators.h"
#include "hash_map.h"
#include "idx.h"
#include "indexed.h"
#include "tensor.h"
#include "color.h"
#include "clifford.h"
#include "factor.h"
#include "integration_kernel.h"
#include "excompiler.h"
#include "Gt.h"
#include "parser.h"
```

9.37.1 Detailed Description

This include file includes all other public GiNaC headers.

9.38 Gt.cpp File Reference

Implementation of GiNaC's elliptic multiple polylogarithms.

```
#include <type_traits>
#include <sstream>
#include <limits>
#include <cln/cln.h>
#include "add.h"
#include "mul.h"
#include "power.h"
#include "lst.h"
#include "matrix.h"
#include "symbol.h"
#include "numeric.h"
#include "inifcns.h"
#include "operators.h"
#include "relational.h"
#include "utils_multi_iterator.h"
#include "hash_seed.h"
#include "constant.h"
#include "Gt.h"
#include "Gt_helpers.h"
```

Namespaces

- namespace GiNaC

Macros

- #define DEBUG_GT 0

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Gt, basic, print_func< print_dflt >(&Gt::do_print). print_func< print_python >(&Gt::do_print). print_func< print_latex >(&Gt::do_print_latex)) Gt
- GiNaC::GINAC_BIND_UNARCHIVER (Gt)

9.38.1 Detailed Description

Implementation of GiNaC's elliptic multiple polylogarithms.

9.38.2 Macro Definition Documentation

9.38.2.1 DEBUG_GT

```
#define DEBUG_GT 0
```

Referenced by GiNaC::Gt::decomposeIntegrationPath(), GiNaC::Gt::ex_evaluate(), GiNaC::Gt::ex_prepare(), GiNaC::Gt::lst_evaluate(), GiNaC::Gt::qExpand(), GiNaC::Gt::regularise(), GiNaC::Gt::tauToFundamental(), and GiNaC::Gt::zisToFundamental().

9.39 Gt.h File Reference

Interface to GiNaC's elliptic multiple polylogarithms.

```
#include "archive.h"
#include "basic.h"
#include "ex.h"
```

Classes

- class GiNaC::Gt
  Elliptic Multiple Polylogarithm.
- struct GiNaC::Gt::kernel

Namespaces

- namespace GiNaC
- namespace GiNaC::Gt_detail

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (Gt)

9.39.1 Detailed Description

Interface to GiNaC's elliptic multiple polylogarithms.

9.40 Gt_helpers.cpp File Reference

Implementation of helper functions for elliptic multiple polylogarithms.

```
#include "add.h"
#include "mul.h"
#include "power.h"
#include "lst.h"
#include "operators.h"
#include "relational.h"
#include "inifcns.h"
#include "function.h"
#include "Gt.h"
#include "Gt_helpers.h"
```

Namespaces

- namespace GiNaC
- namespace GiNaC::Gt_detail

Functions

- std::vector< std::vector< int > > GiNaC::Gt_detail::integer_partition (const int n, const int m)
- template<typename Kernel>
  ex GiNaC::Gt_detail::deconcatenate_path (const std::vector< Kernel > &args, const std::vector< ex > &endpoints, const std::function< ex(std::vector< Kernel > &new_args, const ex &start, const ex &end)> &construct)
- template ex GiNaC::Gt_detail::deconcatenate_path< ex > (const std::vector< ex > &args, const std::vector< ex > &endpoints, const std::function< ex(std::vector< ex > &new_args, const ex &start, const ex &end)> &construct)
- template ex GiNaC::Gt_detail::deconcatenate_path< Gt::kernel > (const std::vector< Gt::kernel > &args, const std::vector< ex > &endpoints, const std::function< ex(std::vector< Gt::kernel > &new_args, const ex &start, const ex &end)> &construct)
- std::ostream & GiNaC::Gt_detail::operator<< (std::ostream &os, const pathintegral_term &term)
- bool GiNaC::Gt_detail::operator< (const pathintegral_term &a, const pathintegral_term &b)

9.40.1 Detailed Description

Implementation of helper functions for elliptic multiple polylogarithms.

9.41 Gt_helpers.h File Reference

Interface to helper functions for elliptic multiple polylogarithms.

```
#include <set>
#include "ex.h"
#include "hash_map.h"
```

Classes

- class GiNaC::Gt_detail::TransformExpressionWithCache< Type >
- class GiNaC::Gt_detail::pathintegral_term
- class GiNaC::Gt_detail::pathintegral_term_list

Namespaces

- namespace GiNaC
- namespace GiNaC::Gt_detail

Functions

- std::vector< std::vector< int > > GiNaC::Gt_detail::integer_partition (const int n, const int m)
- template<typename Kernel>
  ex GiNaC::Gt_detail::deconcatenate_path (const std::vector< Kernel > &args, const std::vector< ex > &endpoints, const std::function< ex(std::vector< Kernel > &new_args, const ex &start, const ex &end)> &construct)
- std::ostream & GiNaC::Gt_detail::operator<< (std::ostream &os, const pathintegral_term &term)
- bool GiNaC::Gt_detail::operator< (const pathintegral_term &a, const pathintegral_term &b)

9.41.1 Detailed Description

Interface to helper functions for elliptic multiple polylogarithms.

9.42 hash_map.h File Reference

Replacement for map<> using hash tables.

```
#include <unordered_map>
```

Namespaces

- namespace GiNaC

Typedefs

- template<typename T, class Hash = std::hash<ex>, class KeyEqual = std::equal_to<ex>, class Allocator = std::allocator<std::pair<const ex, T>>>
  using GiNaC::exhashmap = std::unordered_map<ex, T, Hash, KeyEqual, Allocator>

9.42.1 Detailed Description

Replacement for map<> using hash tables.

9.43 hash_seed.h File Reference

Type-specific hash seed.

```
#include <typeinfo>
#include <cstring>
#include "utils.h"
```

Namespaces

- namespace GiNaC

9.43.1 Detailed Description

Type-specific hash seed.

9.44 idx.cpp File Reference

Implementation of GiNaC's indices.

```
#include "idx.h"
#include "symbol.h"
#include "lst.h"
#include "relational.h"
#include "operators.h"
#include "archive.h"
#include "utils.h"
#include "hash_seed.h"
#include <sstream>
#include <stdexcept>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (idx, basic, print_func< print_context >(&idx::do_print). print_func< print_latex >(&idx::do_print_latex). print_func< print_csrc >(&idx::do_print_csrc). print_func< print_tree >(&idx::do_print_tree)) GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(varidx
- GiNaC::print_func< print_context > (&varidx::do_print). print_func< print_latex >(&varidx
- GiNaC::GINAC_BIND_UNARCHIVER (idx)
- GiNaC::GINAC_BIND_UNARCHIVER (varidx)
- GiNaC::GINAC_BIND_UNARCHIVER (spinidx)
- bool GiNaC::is_dummy_pair (const idx &i1, const idx &i2)

  Check whether two indices form a dummy pair.
- bool GiNaC::is_dummy_pair (const ex &e1, const ex &e2)

  Check whether two expressions form a dummy index pair.
- void GiNaC::find_free_and_dummy (exvector::const_iterator it, exvector::const_iterator itend, exvector &out_free, exvector &out_dummy)

  Given a vector of indices, split them into two vectors, one containing the free indices, the other containing the dummy indices (numeric indices are neither free nor dummy ones).
- ex GiNaC::minimal_dim (const ex &dim1, const ex &dim2)

  Return the minimum of two index dimensions.

Variables

- GiNaC::idx

9.44.1 Detailed Description

Implementation of GiNaC's indices.

9.45 idx.h File Reference

Interface to GiNaC's indices.

```
#include "ex.h"
#include "numeric.h"
```

Classes

- class GiNaC::idx

  This class holds one index of an indexed object.
- class GiNaC::varidx

  This class holds an index with a variance (co- or contravariant).
- class GiNaC::spinidx

  This class holds a spinor index that can be dotted or undotted and that also has a variance.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (idx)
- GiNaC::GINAC_DECLARE_UNARCHIVER (varidx)
- GiNaC::GINAC_DECLARE_UNARCHIVER (spinidx)
- bool GiNaC::is_dummy_pair (const idx &i1, const idx &i2)

  Check whether two indices form a dummy pair.
- bool GiNaC::is_dummy_pair (const ex &e1, const ex &e2)

  Check whether two expressions form a dummy index pair.
- void GiNaC::find_free_and_dummy (exvector::const_iterator it, exvector::const_iterator itend, exvector &out_free, exvector &out_dummy)

  Given a vector of indices, split them into two vectors, one containing the free indices, the other containing the dummy indices (numeric indices are neither free nor dummy ones).
- void GiNaC::find_free_and_dummy (const exvector &v, exvector &out_free, exvector &out_dummy)

  Given a vector of indices, split them into two vectors, one containing the free indices, the other containing the dummy indices (numeric indices are neither free nor dummy ones).
- void GiNaC::find_dummy_indices (const exvector &v, exvector &out_dummy)

  Given a vector of indices, find the dummy indices.
- size_t GiNaC::count_dummy_indices (const exvector &v)

  Count the number of dummy index pairs in an index vector.
- size_t GiNaC::count_free_indices (const exvector &v)

  Count the number of dummy index pairs in an index vector.
- ex GiNaC::minimal_dim (const ex &dim1, const ex &dim2)

  Return the minimum of two index dimensions.

9.45.1 Detailed Description

Interface to GiNaC's indices.

9.46 indexed.cpp File Reference

Implementation of GiNaC's indexed expressions.

```
#include "indexed.h"
#include "idx.h"
#include "add.h"
#include "mul.h"
#include "ncmul.h"
#include "power.h"
#include "relational.h"
#include "symmetry.h"
#include "operators.h"
#include "lst.h"
#include "archive.h"
#include "symbol.h"
#include "utils.h"
#include "integral.h"
#include "matrix.h"
#include "inifcns.h"
#include <algorithm>
#include <iostream>
#include <limits>
#include <sstream>
#include <stdexcept>
```

Classes

- struct GiNaC::idx_is_equal_ignore_dim
- struct GiNaC::is_summation_idx
- struct GiNaC::ex_base_is_less
- class GiNaC::terminfo

  This structure stores the original and symmetrized versions of terms obtained during the simplification of sums.
- class GiNaC::terminfo_is_less
- class GiNaC::symminfo

  This structure stores the individual symmetrized terms obtained during the simplification of sums.
- class GiNaC::symminfo_is_less_by_symmterm
- class GiNaC::symminfo_is_less_by_orig

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (indexed, exprseq, print_func< print_context >(&indexed::do_print). print_func< print_latex >(&indexed::do_print_latex). print_func< print_tree >(&indexed::do_print_tree)) indexed
- GiNaC::GINAC_BIND_UNARCHIVER (indexed)
- static bool GiNaC::indices_consistent (const exvector &v1, const exvector &v2)

  Check whether two sorted index vectors are consistent (i.e.
- template<class T>
  size_t GiNaC::number_of_type (const exvector &v)
- template<class T>
  static ex GiNaC::rename_dummy_indices (const ex &e, exvector &global_dummy_indices, exvector &local↩
  _dummy_indices)

  Rename dummy indices in an expression.
- static void GiNaC::find_variant_indices (const exvector &v, exvector &variant_indices)

  Given a set of indices, extract those of class varidx.
- bool GiNaC::reposition_dummy_indices (ex &e, exvector &variant_dummy_indices, exvector &moved_↩
  indices)

  Raise/lower dummy indices in a single indexed objects to canonicalize their variance.
- static void GiNaC::product_to_exvector (const ex &e, exvector &v, bool &non_commutative)
- template<class T>
  ex GiNaC::idx_symmetrization (const ex &r, const exvector &local_dummy_indices)
- ex GiNaC::simplify_indexed (const ex &e, exvector &free_indices, exvector &dummy_indices, const scalar_products &sp)

  Simplify indexed expression, return list of free indices.
- ex GiNaC::simplify_indexed_product (const ex &e, exvector &free_indices, exvector &dummy_indices, const scalar_products &sp)

  Simplify product of indexed expressions (commutative, noncommutative and simple squares), return list of free indices.
- bool GiNaC::hasindex (const ex &x, const ex &sym)
- exvector GiNaC::get_all_dummy_indices_safely (const ex &e)

  More reliable version of the form.
- exvector GiNaC::get_all_dummy_indices (const ex &e)

  Returns all dummy indices from the exvector.
- lst GiNaC::rename_dummy_indices_uniquely (const exvector &va, const exvector &vb)

  Similar to above, where va and vb are the same and the return value is a list of two lists for substitution in b.
- ex GiNaC::rename_dummy_indices_uniquely (const exvector &va, const exvector &vb, const ex &b)

  Same as above, where va and vb contain the indices of a and b and are sorted.
- ex GiNaC::rename_dummy_indices_uniquely (const ex &a, const ex &b)

  Returns b with all dummy indices, which are common with a, renamed.
- ex GiNaC::rename_dummy_indices_uniquely (exvector &va, const ex &b, bool modify_va=false)

  Returns b with all dummy indices, which are listed in va, renamed if modify_va is set to TRUE all dummy indices of b will be appended to va.
- ex GiNaC::expand_dummy_sum (const ex &e, bool subs_idx=false)

  This function returns the given expression with expanded sums for all dummy index summations, where the dimensionality of the dummy index is a nonnegative integer.

9.46.1 Detailed Description

Implementation of GiNaC's indexed expressions.

9.47 indexed.h File Reference

Interface to GiNaC's indexed expressions.

```
#include "exprseq.h"
#include "wildcard.h"
#include <map>
```

Classes

- class GiNaC::indexed

  This class holds an indexed expression.
- class GiNaC::spmapkey
- class GiNaC::scalar_products

  Helper class for storing information about known scalar products which are to be automatically replaced by simplify_indexed().

Namespaces

- namespace GiNaC

Typedefs

- typedef std::map< spmapkey, ex > GiNaC::spmap

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (indexed)
- exvector GiNaC::get_all_dummy_indices (const ex &e)

  Returns all dummy indices from the exvector.
- exvector GiNaC::get_all_dummy_indices_safely (const ex &e)

  More reliable version of the form.
- ex GiNaC::rename_dummy_indices_uniquely (exvector &va, const ex &b, bool modify_va=false)

  Returns b with all dummy indices, which are listed in va, renamed if modify_va is set to TRUE all dummy indices of b will be appended to va.
- ex GiNaC::rename_dummy_indices_uniquely (const ex &a, const ex &b)

  Returns b with all dummy indices, which are common with a, renamed.
- ex GiNaC::rename_dummy_indices_uniquely (const exvector &va, const exvector &vb, const ex &b)

  Same as above, where va and vb contain the indices of a and b and are sorted.
- lst GiNaC::rename_dummy_indices_uniquely (const exvector &va, const exvector &vb)

  Similar to above, where va and vb are the same and the return value is a list of two lists for substitution in b.
- ex GiNaC::expand_dummy_sum (const ex &e, bool subs_idx=false)

  This function returns the given expression with expanded sums for all dummy index summations, where the dimensionality of the dummy index is a nonnegative integer.

9.47.1 Detailed Description

Interface to GiNaC's indexed expressions.

9.48 inifcns.cpp File Reference

Implementation of GiNaC's initially known functions.

```
#include "inifcns.h"
#include "ex.h"
#include "constant.h"
#include "lst.h"
#include "fderivative.h"
#include "matrix.h"
#include "mul.h"
#include "power.h"
#include "operators.h"
#include "relational.h"
#include "pseries.h"
#include "symbol.h"
#include "symmetry.h"
#include "utils.h"
#include <stdexcept>
#include <vector>
```

Classes

- class GiNaC::symbolset

Namespaces

- namespace GiNaC

Functions

- static ex GiNaC::conjugate_evalf (const ex &arg)
- static ex GiNaC::conjugate_eval (const ex &arg)
- static void GiNaC::conjugate_print_latex (const ex &arg, const print_context &c)
- static ex GiNaC::conjugate_conjugate (const ex &arg)
- static ex GiNaC::conjugate_expl_derivative (const ex &arg, const symbol &s)
- static ex GiNaC::conjugate_real_part (const ex &arg)
- static ex GiNaC::conjugate_imag_part (const ex &arg)
- static bool GiNaC::func_arg_info (const ex &arg, unsigned inf)
- static bool GiNaC::conjugate_info (const ex &arg, unsigned inf)
- GiNaC::REGISTER_FUNCTION (conjugate_function, eval_func(conjugate_eval). evalf_func(conjugate_evalf). expl_derivative_func(conjugate_expl_derivative). info_func(conjugate_info). print_func< print_latex >(conjugate_print_latex). conjugate_func(conjugate_conjugate). real_part_func(conjugate_real_part). imag_part_func(conjugate_imag_part). set_name("conjugate","conjugate"))
- static ex GiNaC::real_part_evalf (const ex &arg)
- static ex GiNaC::real_part_eval (const ex &arg)
- static void GiNaC::real_part_print_latex (const ex &arg, const print_context &c)
- static ex GiNaC::real_part_conjugate (const ex &arg)
- static ex GiNaC::real_part_real_part (const ex &arg)
- static ex GiNaC::real_part_imag_part (const ex &arg)
- static ex GiNaC::real_part_expl_derivative (const ex &arg, const symbol &s)

- GiNaC::REGISTER_FUNCTION (real_part_function, eval_func(real_part_eval). evalf_func(real_part_evalf). expl_derivative_func(real_part_expl_derivative). print_func< print_latex >(real_part_print_latex). conjugate_func(real_part_conjugate). real_part_func(real_part_real_part). imag_part_func(real_part_imag_part). set_name("real_part","real_part"))
- static ex GiNaC::imag_part_evalf (const ex &arg)
- static ex GiNaC::imag_part_eval (const ex &arg)
- static void GiNaC::imag_part_print_latex (const ex &arg, const print_context &c)
- static ex GiNaC::imag_part_conjugate (const ex &arg)
- static ex GiNaC::imag_part_real_part (const ex &arg)
- static ex GiNaC::imag_part_imag_part (const ex &arg)
- static ex GiNaC::imag_part_expl_derivative (const ex &arg, const symbol &s)
- GiNaC::REGISTER_FUNCTION (imag_part_function, eval_func(imag_part_eval). evalf_func(imag_part_evalf). expl_derivative_func(imag_part_expl_derivative). print_func< print_latex >(imag_part_print_latex). conjugate_func(imag_part_conjugate). real_part_func(imag_part_real_part). imag_part_func(imag_part_imag_part). set_name("imag_part","imag_part"))
- static ex GiNaC::abs_evalf (const ex &arg)
- static ex GiNaC::abs_eval (const ex &arg)
- static ex GiNaC::abs_expand (const ex &arg, unsigned options)
- static ex GiNaC::abs_expl_derivative (const ex &arg, const symbol &s)
- static void GiNaC::abs_print_latex (const ex &arg, const print_context &c)
- static void GiNaC::abs_print_csrc_float (const ex &arg, const print_context &c)
- static ex GiNaC::abs_conjugate (const ex &arg)
- static ex GiNaC::abs_real_part (const ex &arg)
- static ex GiNaC::abs_imag_part (const ex &arg)
- static ex GiNaC::abs_power (const ex &arg, const ex &exp)
- bool GiNaC::abs_info (const ex &arg, unsigned inf)
- GiNaC::REGISTER_FUNCTION (abs, eval_func(abs_eval). evalf_func(abs_evalf). expand_func(abs_expand). expl_derivative_func(abs_expl_derivative). info_func(abs_info). print_func< print_latex >(abs_print_latex). print_func< print_csrc_float >(abs_print_csrc_float). print_func< print_csrc_double >(abs_print_csrc_float). conjugate_func(abs_conjugate). real_part_func(abs_real_part). imag_part_func(abs_imag_part). power_func(abs_power))
- static ex GiNaC::step_evalf (const ex &arg)
- static ex GiNaC::step_eval (const ex &arg)
- static ex GiNaC::step_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex GiNaC::step_conjugate (const ex &arg)
- static ex GiNaC::step_real_part (const ex &arg)
- static ex GiNaC::step_imag_part (const ex &arg)
- GiNaC::REGISTER_FUNCTION (step, eval_func(step_eval). evalf_func(step_evalf). series_func(step_series). conjugate_func(step_conjugate). real_part_func(step_real_part). imag_part_func(step_imag_part))
- static ex GiNaC::csgn_evalf (const ex &arg)
- static ex GiNaC::csgn_eval (const ex &arg)
- static ex GiNaC::csgn_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex GiNaC::csgn_conjugate (const ex &arg)
- static ex GiNaC::csgn_real_part (const ex &arg)
- static ex GiNaC::csgn_imag_part (const ex &arg)
- static ex GiNaC::csgn_power (const ex &arg, const ex &exp)
- GiNaC::REGISTER_FUNCTION (csgn, eval_func(csgn_eval). evalf_func(csgn_evalf). series_func(csgn_series). conjugate_func(csgn_conjugate). real_part_func(csgn_real_part). imag_part_func(csgn_imag_part). power_func(csgn_power))
- static ex GiNaC::eta_evalf (const ex &x, const ex &y)
- static ex GiNaC::eta_eval (const ex &x, const ex &y)
- static ex GiNaC::eta_series (const ex &x, const ex &y, const relational &rel, int order, unsigned options)
- static ex GiNaC::eta_conjugate (const ex &x, const ex &y)
- static ex GiNaC::eta_real_part (const ex &x, const ex &y)
- static ex GiNaC::eta_imag_part (const ex &x, const ex &y)

- GiNaC::REGISTER_FUNCTION (eta, eval_func(eta_eval). evalf_func(eta_evalf). series_func(eta_series). latex_name("\\eta"). set_symmetry(sy_symm(0, 1)). conjugate_func(eta_conjugate). real_part_↩ func(eta_real_part). imag_part_func(eta_imag_part))
- static ex GiNaC::Li2_evalf (const ex &x)
- static ex GiNaC::Li2_eval (const ex &x)
- static ex GiNaC::Li2_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::Li2_series (const ex &x, const relational &rel, int order, unsigned options)
- static ex GiNaC::Li2_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (Li2, eval_func(Li2_eval). evalf_func(Li2_evalf). derivative_func(Li2_deriv). series_func(Li2_series). conjugate_func(Li2_conjugate). latex_name("\\mathrm{Li}_2"))
- static ex GiNaC::Li3_eval (const ex &x)
- GiNaC::REGISTER_FUNCTION (Li3, eval_func(Li3_eval). latex_name("\\mathrm{Li}_3"))
- static ex GiNaC::zetaderiv_eval (const ex &n, const ex &x)
- static ex GiNaC::zetaderiv_deriv (const ex &n, const ex &x, unsigned deriv_param)
- GiNaC::REGISTER_FUNCTION (zetaderiv, eval_func(zetaderiv_eval). derivative_func(zetaderiv_deriv). latex_name("\\zeta^\\prime"))
- static ex GiNaC::factorial_evalf (const ex &x)
- static ex GiNaC::factorial_eval (const ex &x)
- static void GiNaC::factorial_print_dflt_latex (const ex &x, const print_context &c)
- static ex GiNaC::factorial_conjugate (const ex &x)
- static ex GiNaC::factorial_real_part (const ex &x)
- static ex GiNaC::factorial_imag_part (const ex &x)
- GiNaC::REGISTER_FUNCTION (factorial, eval_func(factorial_eval). evalf_func(factorial_evalf). print_func< print_dflt >(factorial_print_dflt_latex). print_func< print_latex >(factorial_print_dflt_latex). conjugate_↩ func(factorial_conjugate). real_part_func(factorial_real_part). imag_part_func(factorial_imag_part))
- static ex GiNaC::binomial_evalf (const ex &x, const ex &y)
- static ex GiNaC::binomial_sym (const ex &x, const numeric &y)
- static ex GiNaC::binomial_eval (const ex &x, const ex &y)
- static ex GiNaC::binomial_conjugate (const ex &x, const ex &y)
- static ex GiNaC::binomial_real_part (const ex &x, const ex &y)
- static ex GiNaC::binomial_imag_part (const ex &x, const ex &y)
- GiNaC::REGISTER_FUNCTION (binomial, eval_func(binomial_eval). evalf_func(binomial_evalf). conjugate↩ _func(binomial_conjugate). real_part_func(binomial_real_part). imag_part_func(binomial_imag_part))
- static ex GiNaC::Order_eval (const ex &x)
- static ex GiNaC::Order_series (const ex &x, const relational &r, int order, unsigned options)
- static ex GiNaC::Order_conjugate (const ex &x)
- static ex GiNaC::Order_real_part (const ex &x)
- static ex GiNaC::Order_imag_part (const ex &x)
- static ex GiNaC::Order_power (const ex &x, const ex &e)
- static ex GiNaC::Order_expl_derivative (const ex &arg, const symbol &s)
- GiNaC::REGISTER_FUNCTION (Order, eval_func(Order_eval). series_func(Order_series). latex_↩ name("\\mathcal{O}"). expl_derivative_func(Order_expl_derivative). conjugate_func(Order_conjugate). real_part_func(Order_real_part). imag_part_func(Order_imag_part). power_func(Order_power))
- ex GiNaC::lsolve (const ex &eqns, const ex &symbols, unsigned options=solve_algo::automatic)

  Factorial function.
- const numeric GiNaC::fsolve (const ex &f, const symbol &x, const numeric &x1, const numeric &x2)

  Find a real root of real-valued function f(x) numerically within a given interval.

Variables

- unsigned GiNaC::force_include_tgamma = tgamma_SERIAL::serial
- unsigned GiNaC::force_include_zeta1 = zeta1_SERIAL::serial

9.48.1 Detailed Description

Implementation of GiNaC's initially known functions.

9.49 inifcns.h File Reference

Interface to GiNaC's initially known functions.

```
#include "numeric.h"
#include "function.h"
#include "ex.h"
```

Classes

- class GiNaC::zeta1_SERIAL
  Complex conjugate.
- class GiNaC::zeta2_SERIAL
  Alternating Euler sum or colored MZV.
- class GiNaC::G2_SERIAL
  Generalized multiple polylogarithm.
- class GiNaC::G3_SERIAL
  Generalized multiple polylogarithm with explicit imaginary parts.
- class GiNaC::psi1_SERIAL
  Polylogarithm and multiple polylogarithm.
- class GiNaC::psi2_SERIAL
  Derivatives of Psi-function (aka polygamma-functions).
- class GiNaC::iterated_integral2_SERIAL
  Complete elliptic integral of the first kind.
- class GiNaC::iterated_integral3_SERIAL
  Iterated integral with explicit truncation.

Namespaces

- namespace GiNaC

Functions

- template<typename T1>
  function GiNaC::zeta (const T1 &p1)
- template<typename T1, typename T2>
  function GiNaC::zeta (const T1 &p1, const T2 &p2)
- template<> bool GiNaC::is_the_function< zeta_SERIAL > (const ex &x)
- template<typename T1, typename T2>
  function GiNaC::G (const T1 &x, const T2 &y)
- template<typename T1, typename T2, typename T3>
  function GiNaC::G (const T1 &x, const T2 &s, const T3 &y)
- template<> bool GiNaC::is_the_function< G_SERIAL > (const ex &x)

- template<typename T1>
  function GiNaC::psi (const T1 &p1)
- template<typename T1, typename T2>
  function GiNaC::psi (const T1 &p1, const T2 &p2)
- template<> bool GiNaC::is_the_function< psi_SERIAL > (const ex &x)
- template<typename T1, typename T2>
  function GiNaC::iterated_integral (const T1 &kernel_lst, const T2 &lambda)
- template<typename T1, typename T2, typename T3>
  function GiNaC::iterated_integral (const T1 &kernel_lst, const T2 &lambda, const T3 &N_trunc)
- template<> bool GiNaC::is_the_function< iterated_integral_SERIAL > (const ex &x)
- ex GiNaC::lsolve (const ex &eqns, const ex &symbols, unsigned options=solve_algo::automatic)

  Factorial function.
- const numeric GiNaC::fsolve (const ex &f, const symbol &x, const numeric &x1, const numeric &x2)

  Find a real root of real-valued function f(x) numerically within a given interval.
- bool GiNaC::is_order_function (const ex &e)

  Check whether a function is the Order (O(n)) function.
- ex GiNaC::convert_H_to_Li (const ex &parameterlst, const ex &arg)

  Converts a given list containing parameters for H in Remiddi/Vermaseren notation into the corresponding GiNaC functions.

9.49.1 Detailed Description

Interface to GiNaC's initially known functions.

9.50 inifcns_elliptic.cpp File Reference

Implementation of some special functions related to elliptic curves.

```
#include "inifcns.h"
#include "add.h"
#include "constant.h"
#include "lst.h"
#include "mul.h"
#include "numeric.h"
#include "operators.h"
#include "power.h"
#include "pseries.h"
#include "relational.h"
#include "symbol.h"
#include "utils.h"
#include "wildcard.h"
#include "integration_kernel.h"
#include "utils_multi_iterator.h"
#include <cln/cln.h>
#include <sstream>
#include <stdexcept>
#include <vector>
#include <cmath>
```

Namespaces

- namespace GiNaC

Functions

- static ex GiNaC::EllipticK_evalf (const ex &k)
- static ex GiNaC::EllipticK_eval (const ex &k)
- static ex GiNaC::EllipticK_deriv (const ex &k, unsigned deriv_param)
- static ex GiNaC::EllipticK_series (const ex &k, const relational &rel, int order, unsigned options)
- static void GiNaC::EllipticK_print_latex (const ex &k, const print_context &c)
- GiNaC::REGISTER_FUNCTION (EllipticK, evalf_func(EllipticK_evalf). eval_func(EllipticK_eval). derivative↩ _func(EllipticK_deriv). series_func(EllipticK_series). print_func< print_latex >(EllipticK_print_latex). do_↩ not_evalf_params())
- static ex GiNaC::EllipticE_evalf (const ex &k)
- static ex GiNaC::EllipticE_eval (const ex &k)
- static ex GiNaC::EllipticE_deriv (const ex &k, unsigned deriv_param)
- static ex GiNaC::EllipticE_series (const ex &k, const relational &rel, int order, unsigned options)
- static void GiNaC::EllipticE_print_latex (const ex &k, const print_context &c)
- GiNaC::REGISTER_FUNCTION (EllipticE, evalf_func(EllipticE_evalf). eval_func(EllipticE_eval). derivative↩ _func(EllipticE_deriv). series_func(EllipticE_series). print_func< print_latex >(EllipticE_print_latex). do_↩ not_evalf_params())
- static ex GiNaC::iterated_integral_evalf_impl (const ex &kernel_lst, const ex &lambda, const ex &N_trunc)
- static ex GiNaC::iterated_integral2_evalf (const ex &kernel_lst, const ex &lambda)
- static ex GiNaC::iterated_integral3_evalf (const ex &kernel_lst, const ex &lambda, const ex &N_trunc)
- static ex GiNaC::iterated_integral2_eval (const ex &kernel_lst, const ex &lambda)
- static ex GiNaC::iterated_integral3_eval (const ex &kernel_lst, const ex &lambda, const ex &N_trunc)

9.50.1 Detailed Description

Implementation of some special functions related to elliptic curves.

The functions are: complete elliptic integral of the first kind EllipticK(k) complete elliptic integral of the second kind EllipticE(k) iterated integral iterated_integral(a,y) or iterated_integral(a,y,N_trunc)

Some remarks:

- All formulae used can be looked up in the following publication: [WW] Numerical evaluation of iterated integrals related to elliptic Feynman integrals, M.Walden, S.Weinzierl, arXiv:2010.05271
- When these routines and methods are used for scientific work that leads to publication in a scientific journal, please refer to this program as : M.Walden, S.Weinzierl, "Numerical evaluation of iterated integrals related to elliptic Feynman integrals", arXiv:2010.05271
- As these routines build on the core part of GiNaC, it is also polite to acknowledge C. Bauer, A. Frink, R. Kreckel, "Introduction to the GiNaC Framework for Symbolic Computation within the C++ Programming Language", J. Symbolic Computations 33, 1 (2002), cs.sc/0004015

9.51 inifcns_gamma.cpp File Reference

Implementation of Gamma-function, Beta-function, Polygamma-functions, and some related stuff.

```
#include "inifcns.h"
#include "constant.h"
#include "pseries.h"
#include "numeric.h"
#include "power.h"
#include "relational.h"
#include "operators.h"
#include "symbol.h"
#include "symmetry.h"
#include "utils.h"
#include <stdexcept>
#include <vector>
```

Namespaces

- namespace GiNaC

Functions

- static ex GiNaC::lgamma_evalf (const ex &x)
- static ex GiNaC::lgamma_eval (const ex &x)

  Evaluation of lgamma(x), the natural logarithm of the Gamma function.
- static ex GiNaC::lgamma_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::lgamma_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex GiNaC::lgamma_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (lgamma, eval_func(lgamma_eval). evalf_func(lgamma_evalf). derivative_func(lgamma_deriv). series_func(lgamma_series). conjugate_func(lgamma_conjugate). latex_name("\\log \\Gamma"))
- static ex GiNaC::tgamma_evalf (const ex &x)
- static ex GiNaC::tgamma_eval (const ex &x)

  Evaluation of tgamma(x), the true Gamma function.
- static ex GiNaC::tgamma_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::tgamma_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex GiNaC::tgamma_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (tgamma, eval_func(tgamma_eval). evalf_func(tgamma_evalf). derivative_func(tgamma_deriv). series_func(tgamma_series). conjugate_func(tgamma_conjugate). latex_name("\\Gamma"))
- static ex GiNaC::beta_evalf (const ex &x, const ex &y)
- static ex GiNaC::beta_eval (const ex &x, const ex &y)
- static ex GiNaC::beta_deriv (const ex &x, const ex &y, unsigned deriv_param)
- static ex GiNaC::beta_series (const ex &arg1, const ex &arg2, const relational &rel, int order, unsigned options)
- GiNaC::REGISTER_FUNCTION (beta, eval_func(beta_eval). evalf_func(beta_evalf). derivative_func(beta_deriv). series_func(beta_series). latex_name("\\mathrm{B}"). set_symmetry(sy_symm(0, 1)))
- static ex GiNaC::psi1_evalf (const ex &x)
- static ex GiNaC::psi1_eval (const ex &x)

  Evaluation of digamma-function psi(x).
- static ex GiNaC::psi1_deriv (const ex &x, unsigned deriv_param)

- static ex GiNaC::psi1_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex GiNaC::psi2_evalf (const ex &n, const ex &x)
- static ex GiNaC::psi2_eval (const ex &n, const ex &x)

  Evaluation of polygamma-function psi(n,x).
- static ex GiNaC::psi2_deriv (const ex &n, const ex &x, unsigned deriv_param)
- static ex GiNaC::psi2_series (const ex &n, const ex &arg, const relational &rel, int order, unsigned options)

9.51.1 Detailed Description

Implementation of Gamma-function, Beta-function, Polygamma-functions, and some related stuff.

9.52 inifcns_nstdsums.cpp File Reference

Implementation of some special functions that have a representation as nested sums.

```
#include "inifcns.h"
#include "add.h"
#include "constant.h"
#include "lst.h"
#include "mul.h"
#include "numeric.h"
#include "operators.h"
#include "power.h"
#include "pseries.h"
#include "relational.h"
#include "symbol.h"
#include "utils.h"
#include "wildcard.h"
#include <cln/cln.h>
#include <sstream>
#include <stdexcept>
#include <vector>
#include <cmath>
```

Namespaces

- namespace GiNaC

Functions

- static ex GiNaC::G2_evalf (const ex &x_, const ex &y)
- static ex GiNaC::G2_eval (const ex &x_, const ex &y)
- static ex GiNaC::G3_evalf (const ex &x_, const ex &s_, const ex &y)
- static ex GiNaC::G3_eval (const ex &x_, const ex &s_, const ex &y)
- static ex GiNaC::Li_evalf (const ex &m_, const ex &x_)
- static ex GiNaC::Li_eval (const ex &m_, const ex &x_)
- static ex GiNaC::Li_series (const ex &m, const ex &x, const relational &rel, int order, unsigned options)
- static ex GiNaC::Li_deriv (const ex &m_, const ex &x_, unsigned deriv_param)
- static void GiNaC::Li_print_latex (const ex &m_, const ex &x_, const print_context &c)

- GiNaC::REGISTER_FUNCTION (Li, evalf_func(Li_evalf). eval_func(Li_eval). series_func(Li_series). derivative_func(Li_deriv). print_func< print_latex >(Li_print_latex). do_not_evalf_params())
- static ex GiNaC::S_evalf (const ex &n, const ex &p, const ex &x)
- static ex GiNaC::S_eval (const ex &n, const ex &p, const ex &x)
- static ex GiNaC::S_series (const ex &n, const ex &p, const ex &x, const relational &rel, int order, unsigned options)
- static ex GiNaC::S_deriv (const ex &n, const ex &p, const ex &x, unsigned deriv_param)
- static void GiNaC::S_print_latex (const ex &n, const ex &p, const ex &x, const print_context &c)
- GiNaC::REGISTER_FUNCTION (S, evalf_func(S_evalf). eval_func(S_eval). series_func(S_series). derivative_func(S_deriv). print_func< print_latex >(S_print_latex). do_not_evalf_params())
- static ex GiNaC::H_evalf (const ex &x1, const ex &x2)
- static ex GiNaC::H_eval (const ex &m_, const ex &x)
- static ex GiNaC::H_series (const ex &m, const ex &x, const relational &rel, int order, unsigned options)
- static ex GiNaC::H_deriv (const ex &m_, const ex &x, unsigned deriv_param)
- static void GiNaC::H_print_latex (const ex &m_, const ex &x, const print_context &c)
- GiNaC::REGISTER_FUNCTION (H, evalf_func(H_evalf). eval_func(H_eval). series_func(H_series). derivative_func(H_deriv). print_func< print_latex >(H_print_latex). do_not_evalf_params())
- ex GiNaC::convert_H_to_Li (const ex &parameterlst, const ex &arg)

  Converts a given list containing parameters for H in Remiddi/Vermaseren notation into the corresponding GiNaC functions.
- static ex GiNaC::zeta1_evalf (const ex &x)
- static ex GiNaC::zeta1_eval (const ex &m)
- static ex GiNaC::zeta1_deriv (const ex &m, unsigned deriv_param)
- static void GiNaC::zeta1_print_latex (const ex &m_, const print_context &c)
- static ex GiNaC::zeta2_evalf (const ex &x, const ex &s)
- static ex GiNaC::zeta2_eval (const ex &m, const ex &s_)
- static ex GiNaC::zeta2_deriv (const ex &m, const ex &s, unsigned deriv_param)
- static void GiNaC::zeta2_print_latex (const ex &m_, const ex &s_, const print_context &c)

9.52.1 Detailed Description

Implementation of some special functions that have a representation as nested sums.

The functions are: classical polylogarithm Li(n,x) multiple polylogarithm Li(lst{m_1,...,m_k},lst{x_1,...,x_k}) G(lst{a_1,...,a_k},y) or G(lst{a_1,...,a_k},lst{s_1,...,s_k},y) Nielsen's generalized polylogarithm S(n,p,x) harmonic polylogarithm H(m,x) or H(lst{m_1,...,m_k},x) multiple zeta value zeta(m) or zeta(lst{m_1,...,m_k}) alternating Euler sum zeta(m,s) or zeta(lst{m_1,...,m_k},lst{s_1,...,s_k})

Some remarks:

- All formulae used can be looked up in the following publications: [Kol] Nielsen's Generalized Polylogarithms, K.S.Kolbig, SIAM J.Math.Anal. 17 (1986), pp. 1232-1258. [Cra] Fast Evaluation of Multiple Zeta Sums, R.E.Crandall, Math.Comp. 67 (1998), pp. 1163-1172. [ReV] Harmonic Polylogarithms, E.Remiddi, J.A.M.Vermaseren, Int.J.Mod.Phys. A15 (2000), pp. 725-754 [BBB] Special Values of Multiple Polylogarithms, J.Borwein, D.Bradley, D.Broadhurst, P.Lisonek, Trans.Amer.Math.Soc. 353/3 (2001), pp. 907-941 [VSW] Numerical evaluation of multiple polylogarithms, J.Vollinga, S.Weinzierl, hep-ph/0410259
- The order of parameters and arguments of Li and zeta is defined according to the nested sums representation. The parameters for H are understood as in [ReV]. They can be in expanded — only 0, 1 and -1 — or in compactified — a string with zeros in front of 1 or -1 is written as a single number — notation.
- All functions can be numerically evaluated with arguments in the whole complex plane. The parameters for Li, zeta and S must be positive integers. If you want to have an alternating Euler sum, you have to give the signs of the parameters as a second argument s to zeta(m,s) containing 1 and -1.

- The calculation of classical polylogarithms is speeded up by using Bernoulli numbers and look-up tables. S uses look-up tables as well. The zeta function applies the algorithms in [Cra] and [BBB] for speed up. Multiple polylogarithms use Hoelder convolution [BBB].
- The functions have no means to do a series expansion into nested sums. To do this, you have to convert these functions into the appropriate objects from the nestedsums library, do the expansion and convert the result back.
- Numerical testing of this implementation has been performed by doing a comparison of results between this software and the commercial M..... 4.1. Multiple zeta values have been checked by means of evaluations into simple zeta values. Harmonic polylogarithms have been checked by comparison to S(n,p,x) for corresponding parameter combinations and by continuity checks around |x|=1 along with comparisons to corresponding zeta functions. Multiple polylogarithms were checked against H and zeta and by means of shuffle and quasi-shuffle relations.

9.53 inifcns_trans.cpp File Reference

Implementation of transcendental (and trigonometric and hyperbolic) functions.

```
#include "inifcns.h"
#include "ex.h"
#include "constant.h"
#include "add.h"
#include "mul.h"
#include "numeric.h"
#include "power.h"
#include "operators.h"
#include "relational.h"
#include "symbol.h"
#include "pseries.h"
#include "utils.h"
#include <stdexcept>
#include <vector>
```

Namespaces

- namespace GiNaC

Functions

- static ex GiNaC::exp_evalf (const ex &x)
- static ex GiNaC::exp_eval (const ex &x)
- static ex GiNaC::exp_expand (const ex &arg, unsigned options)
- static ex GiNaC::exp_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::exp_real_part (const ex &x)
- static ex GiNaC::exp_imag_part (const ex &x)
- static ex GiNaC::exp_conjugate (const ex &x)
- static ex GiNaC::exp_power (const ex &x, const ex &a)
- static bool GiNaC::exp_info (const ex &x, unsigned inf)
- GiNaC::REGISTER_FUNCTION (exp, eval_func(exp_eval). evalf_func(exp_evalf). info_func(exp_info). expand_func(exp_expand). derivative_func(exp_deriv). real_part_func(exp_real_part). imag_part_↩ func(exp_imag_part). conjugate_func(exp_conjugate). power_func(exp_power). latex_name("\\exp"))

- static ex GiNaC::log_evalf (const ex &x)
- static ex GiNaC::log_eval (const ex &x)
- static ex GiNaC::log_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::log_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex GiNaC::log_real_part (const ex &x)
- static ex GiNaC::log_imag_part (const ex &x)
- static ex GiNaC::log_expand (const ex &arg, unsigned options)
- static ex GiNaC::log_conjugate (const ex &x)
- static bool GiNaC::log_info (const ex &x, unsigned inf)
- GiNaC::REGISTER_FUNCTION (log, eval_func(log_eval). evalf_func(log_evalf). info_func(log_info). expand_func(log_expand). derivative_func(log_deriv). series_func(log_series). real_part_func(log_real_part). imag_part_func(log_imag_part). conjugate_func(log_conjugate). latex_name("\\ln"))
- static ex GiNaC::sin_evalf (const ex &x)
- static ex GiNaC::sin_eval (const ex &x)
- static ex GiNaC::sin_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::sin_real_part (const ex &x)
- static ex GiNaC::sin_imag_part (const ex &x)
- static ex GiNaC::sin_conjugate (const ex &x)
- static bool GiNaC::trig_info (const ex &x, unsigned inf)
- GiNaC::REGISTER_FUNCTION (sin, eval_func(sin_eval). evalf_func(sin_evalf). info_func(trig_info). derivative_func(sin_deriv). real_part_func(sin_real_part). imag_part_func(sin_imag_part). conjugate_↩ func(sin_conjugate). latex_name("\\sin"))
- static ex GiNaC::cos_evalf (const ex &x)
- static ex GiNaC::cos_eval (const ex &x)
- static ex GiNaC::cos_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::cos_real_part (const ex &x)
- static ex GiNaC::cos_imag_part (const ex &x)
- static ex GiNaC::cos_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (cos, eval_func(cos_eval). info_func(trig_info). evalf_func(cos_evalf). derivative_func(cos_deriv). real_part_func(cos_real_part). imag_part_func(cos_imag_part). conjugate_↩ func(cos_conjugate). latex_name("\\cos"))
- static ex GiNaC::tan_evalf (const ex &x)
- static ex GiNaC::tan_eval (const ex &x)
- static ex GiNaC::tan_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::tan_real_part (const ex &x)
- static ex GiNaC::tan_imag_part (const ex &x)
- static ex GiNaC::tan_series (const ex &x, const relational &rel, int order, unsigned options)
- static ex GiNaC::tan_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (tan, eval_func(tan_eval). evalf_func(tan_evalf). info_func(trig_info). derivative_func(tan_deriv). series_func(tan_series). real_part_func(tan_real_part). imag_part_↩ func(tan_imag_part). conjugate_func(tan_conjugate). latex_name("\\tan"))
- static ex GiNaC::asin_evalf (const ex &x)
- static ex GiNaC::asin_eval (const ex &x)
- static ex GiNaC::asin_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::asin_conjugate (const ex &x)
- static bool GiNaC::asin_info (const ex &x, unsigned inf)
- GiNaC::REGISTER_FUNCTION (asin, eval_func(asin_eval). evalf_func(asin_evalf). info_func(asin_info). derivative_func(asin_deriv). conjugate_func(asin_conjugate). latex_name("\\arcsin"))
- static ex GiNaC::acos_evalf (const ex &x)
- static ex GiNaC::acos_eval (const ex &x)
- static ex GiNaC::acos_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::acos_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (acos, eval_func(acos_eval). evalf_func(acos_evalf). info_func(asin_info). derivative_func(acos_deriv). conjugate_func(acos_conjugate). latex_name("\\arccos"))
- static ex GiNaC::atan_evalf (const ex &x)

- static ex GiNaC::atan_eval (const ex &x)
- static ex GiNaC::atan_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::atan_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex GiNaC::atan_conjugate (const ex &x)
- static bool GiNaC::atan_info (const ex &x, unsigned inf)
- GiNaC::REGISTER_FUNCTION (atan, eval_func(atan_eval). evalf_func(atan_evalf). info_func(atan_info). derivative_func(atan_deriv). series_func(atan_series). conjugate_func(atan_conjugate). latex_↩ name("\\arctan"))
- static ex GiNaC::atan2_evalf (const ex &y, const ex &x)
- static ex GiNaC::atan2_eval (const ex &y, const ex &x)
- static ex GiNaC::atan2_deriv (const ex &y, const ex &x, unsigned deriv_param)
- static bool GiNaC::atan2_info (const ex &y, const ex &x, unsigned inf)
- GiNaC::REGISTER_FUNCTION (atan2, eval_func(atan2_eval). evalf_func(atan2_evalf). info_↩ func(atan2_info). evalf_func(atan2_evalf). derivative_func(atan2_deriv))
- static ex GiNaC::sinh_evalf (const ex &x)
- static ex GiNaC::sinh_eval (const ex &x)
- static ex GiNaC::sinh_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::sinh_real_part (const ex &x)
- static ex GiNaC::sinh_imag_part (const ex &x)
- static ex GiNaC::sinh_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (sinh, eval_func(sinh_eval). evalf_func(sinh_evalf). info_func(atan_info). derivative_func(sinh_deriv). real_part_func(sinh_real_part). imag_part_func(sinh_imag_part). conjugate_↩ func(sinh_conjugate). latex_name("\\sinh"))
- static ex GiNaC::cosh_evalf (const ex &x)
- static ex GiNaC::cosh_eval (const ex &x)
- static ex GiNaC::cosh_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::cosh_real_part (const ex &x)
- static ex GiNaC::cosh_imag_part (const ex &x)
- static ex GiNaC::cosh_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (cosh, eval_func(cosh_eval). evalf_func(cosh_evalf). info_func(exp_info). derivative_func(cosh_deriv). real_part_func(cosh_real_part). imag_part_func(cosh_imag_part). conjugate↩ _func(cosh_conjugate). latex_name("\\cosh"))
- static ex GiNaC::tanh_evalf (const ex &x)
- static ex GiNaC::tanh_eval (const ex &x)
- static ex GiNaC::tanh_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::tanh_series (const ex &x, const relational &rel, int order, unsigned options)
- static ex GiNaC::tanh_real_part (const ex &x)
- static ex GiNaC::tanh_imag_part (const ex &x)
- static ex GiNaC::tanh_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (tanh, eval_func(tanh_eval). evalf_func(tanh_evalf). info_func(atan_info). derivative_func(tanh_deriv). series_func(tanh_series). real_part_func(tanh_real_part). imag_part_↩ func(tanh_imag_part). conjugate_func(tanh_conjugate). latex_name("\\tanh"))
- static ex GiNaC::asinh_evalf (const ex &x)
- static ex GiNaC::asinh_eval (const ex &x)
- static ex GiNaC::asinh_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::asinh_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (asinh, eval_func(asinh_eval). evalf_func(asinh_evalf). info_↩ func(atan_info). derivative_func(asinh_deriv). conjugate_func(asinh_conjugate))
- static ex GiNaC::acosh_evalf (const ex &x)
- static ex GiNaC::acosh_eval (const ex &x)
- static ex GiNaC::acosh_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::acosh_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (acosh, eval_func(acosh_eval). evalf_func(acosh_evalf). info_↩ func(asin_info). derivative_func(acosh_deriv). conjugate_func(acosh_conjugate))
- static ex GiNaC::atanh_evalf (const ex &x)

- static ex GiNaC::atanh_eval (const ex &x)
- static ex GiNaC::atanh_deriv (const ex &x, unsigned deriv_param)
- static ex GiNaC::atanh_series (const ex &arg, const relational &rel, int order, unsigned options)
- static ex GiNaC::atanh_conjugate (const ex &x)
- GiNaC::REGISTER_FUNCTION (atanh, eval_func(atanh_eval). evalf_func(atanh_evalf). info_↩ func(asin_info). derivative_func(atanh_deriv). series_func(atanh_series). conjugate_func(atanh_conjugate))

9.53.1 Detailed Description

Implementation of transcendental (and trigonometric and hyperbolic) functions.

9.54 integral.cpp File Reference

Implementation of GiNaC's symbolic integral.

```
#include "integral.h"
#include "numeric.h"
#include "symbol.h"
#include "add.h"
#include "mul.h"
#include "power.h"
#include "inifcns.h"
#include "wildcard.h"
#include "archive.h"
#include "registrar.h"
#include "utils.h"
#include "operators.h"
#include "relational.h"
```

Classes

- struct GiNaC::error_and_integral
- struct GiNaC::error_and_integral_is_less

Namespaces

- namespace GiNaC

Typedefs

- typedef map< error_and_integral, ex, error_and_integral_is_less > GiNaC::lookup_map

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (integral, basic, print_func< print_dflt >(&integral::do_print). print_func< print_python >(&integral::do_print). print_func< print_latex >(&integral::do_print_latex)) integral
- ex GiNaC::subsvalue (const ex &var, const ex &value, const ex &fun)
- ex GiNaC::adaptivesimpson (const ex &x, const ex &a_in, const ex &b_in, const ex &f, const ex &error)

  Numeric integration routine based upon the "Adaptive Quadrature" one in "Numerical Analysis" by Burden and Faires.
- GiNaC::GINAC_BIND_UNARCHIVER (integral)

9.54.1 Detailed Description

Implementation of GiNaC's symbolic integral.

9.55 integral.h File Reference

Interface to GiNaC's symbolic integral.

```
#include "basic.h"
#include "ex.h"
#include "archive.h"
```

Classes

- class GiNaC::integral
  Symbolic integral.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (integral)
- ex GiNaC::adaptivesimpson (const ex &x, const ex &a_in, const ex &b_in, const ex &f, const ex &error)
  Numeric integration routine based upon the "Adaptive Quadrature" one in "Numerical Analysis" by Burden and Faires.

9.55.1 Detailed Description

Interface to GiNaC's symbolic integral.

9.56 integration_kernel.cpp File Reference

Implementation of GiNaC's integration kernels for iterated integrals.

```
#include "integration_kernel.h"
#include "add.h"
#include "mul.h"
#include "operators.h"
#include "power.h"
#include "relational.h"
#include "symbol.h"
#include "constant.h"
#include "numeric.h"
#include "function.h"
#include "pseries.h"
#include "utils.h"
#include "inifcns.h"
#include <iostream>
#include <stdexcept>
#include <cln/cln.h>
```

Namespaces

- namespace GiNaC

Functions

- ex GiNaC::ifactor (const numeric &n)

  Returns the decomposition of the positive integer n into prime numbers in the form lst(lst(p1,...,pr), lst(a1,...,ar)) such that n = p1^a1 ... pr^ar.
- bool GiNaC::is_discriminant_of_quadratic_number_field (const numeric &n)

  Returns true if the integer n is either one or the discriminant of a quadratic number field.
- numeric GiNaC::kronecker_symbol (const numeric &a, const numeric &n)

  Returns the Kronecker symbol a: integer n: integer.
- numeric GiNaC::primitive_dirichlet_character (const numeric &n, const numeric &a)

  Defines a primitive Dirichlet character through the Kronecker symbol.
- numeric GiNaC::dirichlet_character (const numeric &n, const numeric &a, const numeric &N)

  Defines a Dirichlet character through the Kronecker symbol.
- numeric GiNaC::generalised_Bernoulli_number (const numeric &k, const numeric &b)

  The generalised Bernoulli number.
- ex GiNaC::Bernoulli_polynomial (const numeric &k, const ex &x)

  The Bernoulli polynomials.
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (integration_kernel, basic, print_func< print_context >(&integration_kernel::do_print)) integration_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (integration_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (basic_log_kernel, integration_kernel, print_func< print_context >(&basic_log_kernel::do_print)) basic_log_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (basic_log_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (multiple_polylog_kernel, integration_kernel, print_func< print_context >(&multiple_polylog_kernel::do_print)) multiple_polylog_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (multiple_polylog_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (ELi_kernel, integration_kernel, print_func< print_context >(&ELi_kernel::do_print)) ELi_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (ELi_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Ebar_kernel, integration_kernel, print_func< print_context >(&Ebar_kernel::do_print)) Ebar_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (Ebar_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Kronecker_dtau_kernel, integration_kernel, print_func< print_context >(&Kronecker_dtau_kernel::do_print)) Kronecker_dtau_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (Kronecker_dtau_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Kronecker_dz_kernel, integration_kernel, print_func< print_context >(&Kronecker_dz_kernel::do_print)) Kronecker_dz_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (Kronecker_dz_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Eisenstein_kernel, integration_kernel, print_func< print_context >(&Eisenstein_kernel::do_print)) Eisenstein_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (Eisenstein_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (Eisenstein_h_kernel, integration_kernel, print_func< print_context >(&Eisenstein_h_kernel::do_print)) Eisenstein_h_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (Eisenstein_h_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (modular_form_kernel, integration_kernel, print_func< print_context >(&modular_form_kernel::do_print)) modular_form_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (modular_form_kernel)
- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (user_defined_kernel, integration_kernel, print_func< print_context >(&user_defined_kernel::do_print)) user_defined_kernel
- GiNaC::GINAC_BIND_UNARCHIVER (user_defined_kernel)

9.56.1 Detailed Description

Implementation of GiNaC's integration kernels for iterated integrals.

9.57 integration_kernel.h File Reference

Interface to GiNaC's integration kernels for iterated integrals.

```
#include "basic.h"
#include "archive.h"
#include "numeric.h"
#include "lst.h"
#include <cln/complex.h>
#include <vector>
```

Classes

- class GiNaC::integration_kernel

  The base class for integration kernels for iterated integrals.
- class GiNaC::basic_log_kernel

  The basic integration kernel with a logarithmic singularity at the origin.
- class GiNaC::multiple_polylog_kernel

  The integration kernel for multiple polylogarithms.
- class GiNaC::ELi_kernel

  The ELi-kernel.
- class GiNaC::Ebar_kernel

  The Ebar-kernel.
- class GiNaC::Kronecker_dtau_kernel

  The kernel corresponding to integrating the Kronecker coefficient function $g^{(n)}(z_j, K\tau)$ in τ (or equivalently in $\bar{q}$).
- class GiNaC::Kronecker_dz_kernel

  The kernel corresponding to integrating the Kronecker coefficient function $g^{(n-1)}(z - z_j, K\tau)$ in z.
- class GiNaC::Eisenstein_kernel

  The kernel corresponding to the Eisenstein series $E_{k,N,a,b,K}(\tau)$.
- class GiNaC::Eisenstein_h_kernel

  The kernel corresponding to the Eisenstein series $h_{k,N,r,s}(\tau)$.
- class GiNaC::modular_form_kernel

  A kernel corresponding to a polynomial in Eisenstein series.
- class GiNaC::user_defined_kernel

  A user-defined integration kernel.

Namespaces

- namespace GiNaC

Functions

- ex GiNaC::ifactor (const numeric &n)

  Returns the decomposition of the positive integer n into prime numbers in the form lst(lst(p1,...,pr), lst(a1,...,ar)) such that n = p1^a1 ... pr^ar.
- bool GiNaC::is_discriminant_of_quadratic_number_field (const numeric &n)

  Returns true if the integer n is either one or the discriminant of a quadratic number field.
- numeric GiNaC::kronecker_symbol (const numeric &a, const numeric &n)

  Returns the Kronecker symbol a: integer n: integer.
- numeric GiNaC::primitive_dirichlet_character (const numeric &n, const numeric &a)

  Defines a primitive Dirichlet character through the Kronecker symbol.
- numeric GiNaC::dirichlet_character (const numeric &n, const numeric &a, const numeric &N)

  Defines a Dirichlet character through the Kronecker symbol.
- numeric GiNaC::generalised_Bernoulli_number (const numeric &k, const numeric &b)

  The generalised Bernoulli number.
- ex GiNaC::Bernoulli_polynomial (const numeric &k, const ex &x)

  The Bernoulli polynomials.
- GiNaC::GINAC_DECLARE_UNARCHIVER (integration_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (basic_log_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (multiple_polylog_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (ELi_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (Ebar_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (Kronecker_dtau_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (Kronecker_dz_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (Eisenstein_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (Eisenstein_h_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (modular_form_kernel)
- GiNaC::GINAC_DECLARE_UNARCHIVER (user_defined_kernel)

9.57.1 Detailed Description

Interface to GiNaC's integration kernels for iterated integrals.

9.58 lst.cpp File Reference

Implementation of GiNaC's lst.

```
#include "lst.h"
#include "archive.h"
```

Namespaces

- namespace GiNaC

Variables

- template<> GINAC_IMPLEMENT_REGISTERED_CLASS_OPT_T(lst, basic, print_func< print_context >(&lst::do_print). print_func< print_tree >(&lst::do_print_tree)) template<> bool lst GiNaC::GINAC_BIND_UNARCHIVER (lst)

  Specialization of container::info() for lst.

9.58.1 Detailed Description

Implementation of GiNaC's lst.

9.59 lst.h File Reference

Definition of GiNaC's lst.

```
#include "container.h"
#include <list>
```

Namespaces

- namespace GiNaC

Typedefs

- typedef container< std::list > GiNaC::lst

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (lst)

9.59.1 Detailed Description

Definition of GiNaC's lst.

9.60 matrix.cpp File Reference

Implementation of symbolic matrices.

```
#include "matrix.h"
#include "numeric.h"
#include "lst.h"
#include "idx.h"
#include "indexed.h"
#include "add.h"
#include "power.h"
#include "symbol.h"
#include "operators.h"
#include "normal.h"
#include "archive.h"
#include "utils.h"
#include <map>
#include <sstream>
#include <stdexcept>
#include <string>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (matrix, basic, print_func< print_context >(&matrix::do_print). print_func< print_latex >(&matrix::do_print_latex). print_func< print_tree >(&matrix::do_print_tree). print_func< print_python_repr >(&matrix::do_print_python_repr)) matrix

  Default ctor.
- GiNaC::GINAC_BIND_UNARCHIVER (matrix)
- ex GiNaC::lst_to_matrix (const lst &l)

  Convert list of lists to matrix.
- ex GiNaC::diag_matrix (const lst &l)

  Convert list of diagonal elements to matrix.
- ex GiNaC::diag_matrix (std::initializer_list< ex > l)
- ex GiNaC::unit_matrix (unsigned r, unsigned c)

  Create an r times c unit matrix.
- ex GiNaC::symbolic_matrix (unsigned r, unsigned c, const std::string &base_name, const std::string &tex_↩ base_name)

  Create an r times c matrix of newly generated symbols consisting of the given base name plus the numeric row/column position of each element.
- ex GiNaC::reduced_matrix (const matrix &m, unsigned r, unsigned c)

  Return the reduced matrix that is formed by deleting the rth row and cth column of matrix m.
- ex GiNaC::sub_matrix (const matrix &m, unsigned r, unsigned nr, unsigned c, unsigned nc)

  Return the nr times nc submatrix starting at position r, c of matrix m.

9.60.1 Detailed Description

Implementation of symbolic matrices.

9.61 matrix.h File Reference

Interface to symbolic matrices.

```
#include "basic.h"
#include "ex.h"
#include "archive.h"
#include "compiler.h"
#include <string>
#include <vector>
```

Classes

- class GiNaC::matrix

  Symbolic matrices.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (matrix)
- size_t GiNaC::nops (const matrix &m)
- ex GiNaC::expand (const matrix &m, unsigned options=0)
- ex GiNaC::evalf (const matrix &m)
- unsigned GiNaC::rows (const matrix &m)
- unsigned GiNaC::cols (const matrix &m)
- matrix GiNaC::transpose (const matrix &m)
- ex GiNaC::determinant (const matrix &m, unsigned options=determinant_algo::automatic)
- ex GiNaC::trace (const matrix &m)
- ex GiNaC::charpoly (const matrix &m, const ex &lambda)
- matrix GiNaC::inverse (const matrix &m)
- matrix GiNaC::inverse (const matrix &m, unsigned algo)
- unsigned GiNaC::rank (const matrix &m)
- unsigned GiNaC::rank (const matrix &m, unsigned solve_algo)
- ex GiNaC::lst_to_matrix (const lst &l)

  Convert list of lists to matrix.
- ex GiNaC::diag_matrix (const lst &l)

  Convert list of diagonal elements to matrix.
- ex GiNaC::diag_matrix (std::initializer_list< ex > l)
- ex GiNaC::unit_matrix (unsigned r, unsigned c)

  Create an r times c unit matrix.
- ex GiNaC::unit_matrix (unsigned x)

  Create a x times x unit matrix.
- ex GiNaC::symbolic_matrix (unsigned r, unsigned c, const std::string &base_name, const std::string &tex_↩ base_name)

  Create an r times c matrix of newly generated symbols consisting of the given base name plus the numeric row/column position of each element.
- ex GiNaC::reduced_matrix (const matrix &m, unsigned r, unsigned c)

  Return the reduced matrix that is formed by deleting the rth row and cth column of matrix m.
- ex GiNaC::sub_matrix (const matrix &m, unsigned r, unsigned nr, unsigned c, unsigned nc)

  Return the nr times nc submatrix starting at position r, c of matrix m.
- ex GiNaC::symbolic_matrix (unsigned r, unsigned c, const std::string &base_name)

  Create an r times c matrix of newly generated symbols consisting of the given base name plus the numeric row/column position of each element.

9.61.1 Detailed Description

Interface to symbolic matrices.

9.62 mul.cpp File Reference

Implementation of GiNaC's products of expressions.

```
#include "mul.h"
#include "add.h"
#include "power.h"
#include "operators.h"
#include "matrix.h"
#include "indexed.h"
#include "lst.h"
#include "archive.h"
#include "utils.h"
#include "symbol.h"
#include "compiler.h"
#include <limits>
#include <stdexcept>
#include <vector>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (mul, expairseq, print_func< print_context >(&mul::do_print). print_func< print_latex >(&mul::do_print_latex). print_func< print_csrc >(&mul::do_print_csrc). print_func< print_tree >(&mul::do_print_tree). print_func< print_python_repr >(&mul::do_print_python_repr)) mul
- bool GiNaC::tryfactsubs (const ex &origfactor, const ex &patternfactor, int &nummatches, exmap &repls)
- bool GiNaC::algebraic_match_mul_with_mul (const mul &e, const ex &pat, exmap &repls, int factor, int &nummatches, const std::vector< bool > &subsed, std::vector< bool > &matched)

  Checks whether e matches to the pattern pat and the (possibly to be updated) list of replacements repls.
- GiNaC::GINAC_BIND_UNARCHIVER (mul)

9.62.1 Detailed Description

Implementation of GiNaC's products of expressions.

9.63 mul.h File Reference

Interface to GiNaC's products of expressions.

```
#include "expairseq.h"
```

Classes

- class GiNaC::mul

  Product of expressions.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (mul)

9.63.1 Detailed Description

Interface to GiNaC's products of expressions.

9.64 ncmul.cpp File Reference

Implementation of GiNaC's non-commutative products of expressions.

```
#include "ncmul.h"
#include "ex.h"
#include "add.h"
#include "mul.h"
#include "clifford.h"
#include "matrix.h"
#include "archive.h"
#include "indexed.h"
#include "utils.h"
#include <algorithm>
#include <stdexcept>
```

Namespaces

- namespace GiNaC

Typedefs

- typedef std::vector< std::size_t > GiNaC::uintvector
- typedef std::vector< unsigned > GiNaC::unsignedvector
- typedef std::vector< exvector > GiNaC::exvectorvector

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (ncmul, exprseq, print_func< print_context >(&ncmul::do_print). print_func< print_tree >(&ncmul::do_print_tree). print_func< print_csrc >(&ncmul::do_print_csrc). print_func< print_python_repr >(&ncmul::do_print_csrc)) ncmul
- ex GiNaC::reeval_ncmul (const exvector &v)
- ex GiNaC::hold_ncmul (const exvector &v)
- GiNaC::GINAC_BIND_UNARCHIVER (ncmul)

9.64.1 Detailed Description

Implementation of GiNaC's non-commutative products of expressions.

9.65 ncmul.h File Reference

Interface to GiNaC's non-commutative products of expressions.

```
#include "exprseq.h"
#include "archive.h"
```

Classes

- class GiNaC::ncmul

  Non-commutative product of expressions.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (ncmul)
- ex GiNaC::reeval_ncmul (const exvector &v)
- ex GiNaC::hold_ncmul (const exvector &v)

9.65.1 Detailed Description

Interface to GiNaC's non-commutative products of expressions.

9.66 normal.cpp File Reference

This file implements several functions that work on univariate and multivariate polynomials and rational functions.

```
#include "normal.h"
#include "basic.h"
#include "ex.h"
#include "add.h"
#include "constant.h"
#include "expairseq.h"
#include "fail.h"
#include "inifcns.h"
#include "lst.h"
#include "mul.h"
#include "numeric.h"
#include "power.h"
#include "relational.h"
#include "operators.h"
#include "matrix.h"
#include "pseries.h"
#include "symbol.h"
#include "utils.h"
#include "polynomial/chinrem_gcd.h"
#include <algorithm>
#include <map>
```

Classes

- struct GiNaC::sym_desc

  This structure holds information about the highest and lowest degrees in which a symbol appears in two multivariate polynomials "a" and "b".
- class GiNaC::gcdheu_failed

  Exception thrown by heur_gcd() to signal failure.
- struct GiNaC::normal_map_function

  Function object to be applied by basic::normal().

Namespaces

- namespace GiNaC

Macros

- #define FAST_COMPARE 1
- #define USE_REMEMBER 0
- #define USE_TRIAL_DIVISION 0
- #define STATISTICS 0

Typedefs

- typedef std::vector< sym_desc > GiNaC::sym_desc_vec

Functions

- static bool GiNaC::get_first_symbol (const ex &e, ex &x)

  Return pointer to first symbol found in expression.
- static void GiNaC::add_symbol (const ex &s, sym_desc_vec &v)
- static void GiNaC::collect_symbols (const ex &e, sym_desc_vec &v)
- static void GiNaC::get_symbol_stats (const ex &a, const ex &b, sym_desc_vec &v)

  Collect statistical information about symbols in polynomials.
- static numeric GiNaC::lcmcoeff (const ex &e, const numeric &l)
- static numeric GiNaC::lcm_of_coefficients_denominators (const ex &e)

  Compute LCM of denominators of coefficients of a polynomial.
- static ex GiNaC::multiply_lcm (const ex &e, const numeric &lcm)

  Bring polynomial from Q[X] to Z[X] by multiplying in the previously determined LCM of the coefficient's denominators.
- ex GiNaC::quo (const ex &a, const ex &b, const ex &x, bool check_args)

  Quotient q(x) of polynomials a(x) and b(x) in Q[x].
- ex GiNaC::rem (const ex &a, const ex &b, const ex &x, bool check_args)

  Remainder r(x) of polynomials a(x) and b(x) in Q[x].
- ex GiNaC::decomp_rational (const ex &a, const ex &x)

  Decompose rational function a(x)=N(x)/D(x) into P(x)+n(x)/D(x) with degree(n, x) < degree(D, x).
- ex GiNaC::prem (const ex &a, const ex &b, const ex &x, bool check_args)

  Pseudo-remainder of polynomials a(x) and b(x) in Q[x].
- ex GiNaC::sprem (const ex &a, const ex &b, const ex &x, bool check_args)

  Sparse pseudo-remainder of polynomials a(x) and b(x) in Q[x].
- bool GiNaC::divide (const ex &a, const ex &b, ex &q, bool check_args)

  Exact polynomial division of a(X) by b(X) in Q[X].
- static bool GiNaC::divide_in_z (const ex &a, const ex &b, ex &q, sym_desc_vec::const_iterator var)

  Exact polynomial division of a(X) by b(X) in Z[X].
- static ex GiNaC::sr_gcd (const ex &a, const ex &b, sym_desc_vec::const_iterator var)

  Compute GCD of multivariate polynomials using the subresultant PRS algorithm.
- static ex GiNaC::interpolate (const ex &gamma, const numeric &xi, const ex &x, int degree_hint=1)

  xi-adic polynomial interpolation
- static bool GiNaC::heur_gcd_z (ex &res, const ex &a, const ex &b, ex ∗ca, ex ∗cb, sym_desc_vec::const_↩ iterator var)

  Compute GCD of multivariate polynomials using the heuristic GCD algorithm.
- static bool GiNaC::heur_gcd (ex &res, const ex &a, const ex &b, ex ∗ca, ex ∗cb, sym_desc_vec::const_iterator var)

  Compute GCD of multivariate polynomials using the heuristic GCD algorithm.
- static ex GiNaC::gcd_pf_pow (const ex &a, const ex &b, ex ∗ca, ex ∗cb)
- static ex GiNaC::gcd_pf_mul (const ex &a, const ex &b, ex ∗ca, ex ∗cb)
- ex GiNaC::gcd (const ex &a, const ex &b, ex ∗ca, ex ∗cb, bool check_args, unsigned options)

  Compute GCD (Greatest Common Divisor) of multivariate polynomials a(X) and b(X) in Z[X].
- static ex GiNaC::gcd_pf_pow_pow (const ex &a, const ex &b, ex ∗ca, ex ∗cb)
- ex GiNaC::lcm (const ex &a, const ex &b, bool check_args)

  Compute LCM (Least Common Multiple) of multivariate polynomials in Z[X].
- static epvector GiNaC::sqrfree_yun (const ex &a, const symbol &x)

  Compute square-free factorization of multivariate polynomial a(x) using Yun's algorithm.
- ex GiNaC::sqrfree (const ex &a, const lst &l)

  Compute a square-free factorization of a multivariate polynomial in Q[X].
- ex GiNaC::sqrfree_parfrac (const ex &a, const symbol &x)

  Compute square-free partial fraction decomposition of rational function a(x).
- static ex GiNaC::replace_with_symbol (const ex &e, exmap &repl, exmap &rev_lookup, lst &modifier)

  Create a symbol for replacing the expression "e" (or return a previously assigned symbol).

- static ex GiNaC::replace_with_symbol (const ex &e, exmap &repl)

  Create a symbol for replacing the expression "e" (or return a previously assigned symbol).
- static ex GiNaC::frac_cancel (const ex &n, const ex &d)

  Fraction cancellation.
- static ex GiNaC::find_common_factor (const ex &e, ex &factor, exmap &repl)

  Remove the common factor in the terms of a sum 'e' by calculating the GCD, and multiply it into the expression 'factor' (which needs to be initialized to 1, unless you're accumulating factors).
- ex GiNaC::collect_common_factors (const ex &e)

  Collect common factors in sums.
- ex GiNaC::resultant (const ex &e1, const ex &e2, const ex &s)

  Resultant of two expressions e1,e2 with respect to symbol s.

9.66.1 Detailed Description

This file implements several functions that work on univariate and multivariate polynomials and rational functions.

These functions include polynomial quotient and remainder, GCD and LCM computation, square-free factorization and rational function normalization.

9.66.2 Macro Definition Documentation

9.66.2.1 FAST_COMPARE

```
#define FAST_COMPARE 1
```

9.66.2.2 USE_REMEMBER

```
#define USE_REMEMBER 0
```

9.66.2.3 USE_TRIAL_DIVISION

```
#define USE_TRIAL_DIVISION 0
```

9.66.2.4 STATISTICS

```
#define STATISTICS 0
```

9.67 normal.h File Reference

This file defines several functions that work on univariate and multivariate polynomials and rational functions.

```
#include "lst.h"
```

Classes

- struct GiNaC::gcd_options

  Flags to control the behavior of gcd() and friends.

Namespaces

- namespace GiNaC

Functions

- ex GiNaC::quo (const ex &a, const ex &b, const ex &x, bool check_args)

  Quotient q(x) of polynomials a(x) and b(x) in Q[x].
- ex GiNaC::rem (const ex &a, const ex &b, const ex &x, bool check_args)

  Remainder r(x) of polynomials a(x) and b(x) in Q[x].
- ex GiNaC::decomp_rational (const ex &a, const ex &x)

  Decompose rational function a(x)=N(x)/D(x) into P(x)+n(x)/D(x) with degree(n, x) $<$ degree(D, x).
- ex GiNaC::prem (const ex &a, const ex &b, const ex &x, bool check_args)

  Pseudo-remainder of polynomials a(x) and b(x) in Q[x].
- ex GiNaC::sprem (const ex &a, const ex &b, const ex &x, bool check_args)

  Sparse pseudo-remainder of polynomials a(x) and b(x) in Q[x].
- bool GiNaC::divide (const ex &a, const ex &b, ex &q, bool check_args)

  Exact polynomial division of a(X) by b(X) in Q[X].
- ex GiNaC::gcd (const ex &a, const ex &b, ex $*$ca, ex $*$cb, bool check_args, unsigned options)

  Compute GCD (Greatest Common Divisor) of multivariate polynomials a(X) and b(X) in Z[X].
- ex GiNaC::lcm (const ex &a, const ex &b, bool check_args)

  Compute LCM (Least Common Multiple) of multivariate polynomials in Z[X].
- ex GiNaC::sqrfree (const ex &a, const lst &l)

  Compute a square-free factorization of a multivariate polynomial in Q[X].
- ex GiNaC::sqrfree_parfrac (const ex &a, const symbol &x)

  Compute square-free partial fraction decomposition of rational function a(x).
- ex GiNaC::collect_common_factors (const ex &e)

  Collect common factors in sums.
- ex GiNaC::resultant (const ex &e1, const ex &e2, const ex &s)

  Resultant of two expressions e1,e2 with respect to symbol s.

9.67.1 Detailed Description

This file defines several functions that work on univariate and multivariate polynomials and rational functions.

These functions include polynomial quotient and remainder, GCD and LCM computation, square-free factorization and rational function normalization.

9.68 numeric.cpp File Reference

This file contains the interface to the underlying bignum package.

```
#include "numeric.h"
#include "ex.h"
#include "operators.h"
#include "archive.h"
#include "utils.h"
#include <limits>
#include <sstream>
#include <stdexcept>
#include <string>
#include <vector>
#include <cln/output.h>
#include <cln/integer_io.h>
#include <cln/integer_ring.h>
#include <cln/rational_io.h>
#include <cln/rational_ring.h>
#include <cln/lfloat_class.h>
#include <cln/lfloat_io.h>
#include <cln/real_io.h>
#include <cln/real_ring.h>
#include <cln/complex_io.h>
#include <cln/complex_ring.h>
#include <cln/numtheory.h>
```

Classes

- class GiNaC::lanczos_coeffs

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (numeric, basic, print_func< print_context >(&numeric::do_print). print_func< print_latex >(&numeric::do_print_latex). print_func< print_csrc >(&numeric::do_print_csrc). print_func< print_csrc_cl_N >(&numeric::do_print_csrc_cl_N). print_func< print_tree >(&numeric::do_print_tree). print_func< print_python_repr >(&numeric::do_print_python_repr)) numeric

  default ctor.
- static const cln::cl_F GiNaC::make_real_float (const cln::cl_idecoded_float &dec)

  Construct a floating point number from sign, mantissa, and exponent.
- static const cln::cl_F GiNaC::read_real_float (std::istream &s)

  Read serialized floating point number.
- GiNaC::GINAC_BIND_UNARCHIVER (numeric)
- static void GiNaC::write_real_float (std::ostream &s, const cln::cl_R &n)
- static void GiNaC::print_real_number (const print_context &c, const cln::cl_R &x)

  Helper function to print a real number in a nicer way than is CLN's default.
- static void GiNaC::print_integer_csrc (const print_context &c, const cln::cl_I &x)

Helper function to print integer number in C++ source format.

- static void GiNaC::print_real_csrc (const print_context &c, const cln::cl_R &x)

  Helper function to print real number in C++ source format.
- template<typename T1, typename T2>
  static bool GiNaC::coerce (T1 &dst, const T2 &arg)
- template<> bool GiNaC::coerce< int, cln::cl_I > (int &dst, const cln::cl_I &arg)

  Check if CLN integer can be converted into int.
- template<> bool GiNaC::coerce< unsigned int, cln::cl_I > (unsigned int &dst, const cln::cl_I &arg)
- static void GiNaC::print_real_cl_N (const print_context &c, const cln::cl_R &x)

  Helper function to print real number in C++ source format using cl_N types.
- const numeric GiNaC::exp (const numeric &x)

  Exponential function.
- const numeric GiNaC::log (const numeric &x)

  Natural logarithm.
- const numeric GiNaC::sin (const numeric &x)

  Numeric sine (trigonometric function).
- const numeric GiNaC::cos (const numeric &x)

  Numeric cosine (trigonometric function).
- const numeric GiNaC::tan (const numeric &x)

  Numeric tangent (trigonometric function).
- const numeric GiNaC::asin (const numeric &x)

  Numeric inverse sine (trigonometric function).
- const numeric GiNaC::acos (const numeric &x)

  Numeric inverse cosine (trigonometric function).
- const numeric GiNaC::atan (const numeric &x)

  Numeric arcustangent.
- const numeric GiNaC::atan (const numeric &y, const numeric &x)

  Numeric arcustangent of two arguments, analytically continued in a suitable way.
- const numeric GiNaC::sinh (const numeric &x)

  Numeric hyperbolic sine (trigonometric function).
- const numeric GiNaC::cosh (const numeric &x)

  Numeric hyperbolic cosine (trigonometric function).
- const numeric GiNaC::tanh (const numeric &x)

  Numeric hyperbolic tangent (trigonometric function).
- const numeric GiNaC::asinh (const numeric &x)

  Numeric inverse hyperbolic sine (trigonometric function).
- const numeric GiNaC::acosh (const numeric &x)

  Numeric inverse hyperbolic cosine (trigonometric function).
- const numeric GiNaC::atanh (const numeric &x)

  Numeric inverse hyperbolic tangent (trigonometric function).
- static cln::cl_N GiNaC::Li2_series (const cln::cl_N &x, const cln::float_format_t &prec)

  Numeric evaluation of Dilogarithm within circle of convergence (unit circle) using a power series.
- static cln::cl_N GiNaC::Li2_projection (const cln::cl_N &x, const cln::float_format_t &prec)

  Folds Li2's argument inside a small rectangle to enhance convergence.
- const cln::cl_N GiNaC::Li2_ (const cln::cl_N &value)

  Numeric evaluation of Dilogarithm.
- const numeric GiNaC::Li2 (const numeric &x)
- const numeric GiNaC::zeta (const numeric &x)

  Numeric evaluation of Riemann's Zeta function.
- static cln::float_format_t GiNaC::guess_precision (const cln::cl_N &x)
- const cln::cl_N GiNaC::lgamma (const cln::cl_N &x)

The Gamma function.

- const numeric GiNaC::lgamma (const numeric &x)
- const cln::cl_N GiNaC::tgamma (const cln::cl_N &x)
- const numeric GiNaC::tgamma (const numeric &x)
- const numeric GiNaC::psi (const numeric &x)

  The psi function (aka polygamma function).
- const numeric GiNaC::psi (const numeric &n, const numeric &x)

  The psi functions (aka polygamma functions).
- const numeric GiNaC::factorial (const numeric &n)

  Factorial combinatorial function.
- const numeric GiNaC::doublefactorial (const numeric &n)

  The double factorial combinatorial function.
- const numeric GiNaC::binomial (const numeric &n, const numeric &k)

  The Binomial coefficients.
- const numeric GiNaC::bernoulli (const numeric &nn)

  Bernoulli number.
- const numeric GiNaC::fibonacci (const numeric &n)

  Fibonacci number.
- const numeric GiNaC::abs (const numeric &x)

  Absolute value.
- const numeric GiNaC::mod (const numeric &a, const numeric &b)

  Modulus (in positive representation).
- const numeric GiNaC::smod (const numeric &a_, const numeric &b_)

  Modulus (in symmetric representation).
- const numeric GiNaC::irem (const numeric &a, const numeric &b)

  Numeric integer remainder.
- const numeric GiNaC::irem (const numeric &a, const numeric &b, numeric &q)

  Numeric integer remainder.
- const numeric GiNaC::iquo (const numeric &a, const numeric &b)

  Numeric integer quotient.
- const numeric GiNaC::iquo (const numeric &a, const numeric &b, numeric &r)

  Numeric integer quotient.
- const numeric GiNaC::gcd (const numeric &a, const numeric &b)

  Greatest Common Divisor.
- const numeric GiNaC::lcm (const numeric &a, const numeric &b)

  Least Common Multiple.
- const numeric GiNaC::sqrt (const numeric &x)

  Numeric square root.
- const numeric GiNaC::isqrt (const numeric &x)

  Integer numeric square root.
- ex GiNaC::PiEvalf ()

  Floating point evaluation of Archimedes' constant Pi.
- ex GiNaC::EulerEvalf ()

  Floating point evaluation of Euler's constant gamma.
- ex GiNaC::CatalanEvalf ()

  Floating point evaluation of Catalan's constant.
- std::ostream & GiNaC::operator<< (std::ostream &os, const _numeric_digits &e)

Variables

- const numeric GiNaC::I = numeric(cln::complex(cln::cl_I(0),cln::cl_I(1)))

  Imaginary unit.
- _numeric_digits GiNaC::Digits

  Accuracy in decimal digits.

9.68.1 Detailed Description

This file contains the interface to the underlying bignum package.

Its most important design principle is to completely hide the inner working of that other package from the user of GiNaC. It must either provide implementation of arithmetic operators and numerical evaluation of special functions or implement the interface to the bignum package.

9.69 numeric.h File Reference

Makes the interface to the underlying bignum package available.

```
#include "basic.h"
#include "ex.h"
#include "archive.h"
#include <cln/complex.h>
#include <stdexcept>
#include <vector>
```

Classes

- class GiNaC::_numeric_digits

  This class is used to instantiate a global singleton object Digits which behaves just like Maple's Digits.
- class GiNaC::pole_error

  Exception class thrown when a singularity is encountered.
- class GiNaC::numeric

  This class is a wrapper around CLN-numbers within the GiNaC class hierarchy.

Namespaces

- namespace GiNaC

Typedefs

- typedef void(∗) GiNaC::digits_changed_callback(long)

  Function pointer to implement callbacks in the case 'Digits' gets changed.

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (numeric)
- const numeric GiNaC::exp (const numeric &x)

  Exponential function.
- const numeric GiNaC::log (const numeric &x)

  Natural logarithm.
- const numeric GiNaC::sin (const numeric &x)

  Numeric sine (trigonometric function).
- const numeric GiNaC::cos (const numeric &x)

  Numeric cosine (trigonometric function).
- const numeric GiNaC::tan (const numeric &x)

  Numeric tangent (trigonometric function).
- const numeric GiNaC::asin (const numeric &x)

  Numeric inverse sine (trigonometric function).
- const numeric GiNaC::acos (const numeric &x)

  Numeric inverse cosine (trigonometric function).
- const numeric GiNaC::atan (const numeric &x)

  Numeric arcustangent.
- const numeric GiNaC::atan (const numeric &y, const numeric &x)

  Numeric arcustangent of two arguments, analytically continued in a suitable way.
- const numeric GiNaC::sinh (const numeric &x)

  Numeric hyperbolic sine (trigonometric function).
- const numeric GiNaC::cosh (const numeric &x)

  Numeric hyperbolic cosine (trigonometric function).
- const numeric GiNaC::tanh (const numeric &x)

  Numeric hyperbolic tangent (trigonometric function).
- const numeric GiNaC::asinh (const numeric &x)

  Numeric inverse hyperbolic sine (trigonometric function).
- const numeric GiNaC::acosh (const numeric &x)

  Numeric inverse hyperbolic cosine (trigonometric function).
- const numeric GiNaC::atanh (const numeric &x)

  Numeric inverse hyperbolic tangent (trigonometric function).
- const numeric GiNaC::Li2 (const numeric &x)
- const numeric GiNaC::zeta (const numeric &x)

  Numeric evaluation of Riemann's Zeta function.
- const numeric GiNaC::lgamma (const numeric &x)
- const numeric GiNaC::tgamma (const numeric &x)
- const numeric GiNaC::psi (const numeric &x)

  The psi function (aka polygamma function).
- const numeric GiNaC::psi (const numeric &n, const numeric &x)

  The psi functions (aka polygamma functions).
- const numeric GiNaC::factorial (const numeric &n)

  Factorial combinatorial function.
- const numeric GiNaC::doublefactorial (const numeric &n)

  The double factorial combinatorial function.
- const numeric GiNaC::binomial (const numeric &n, const numeric &k)

  The Binomial coefficients.
- const numeric GiNaC::bernoulli (const numeric &nn)

  Bernoulli number.
- const numeric GiNaC::fibonacci (const numeric &n)

Fibonacci number.

- const numeric GiNaC::isqrt (const numeric &x)

  Integer numeric square root.
- const numeric GiNaC::sqrt (const numeric &x)

  Numeric square root.
- const numeric GiNaC::abs (const numeric &x)

  Absolute value.
- const numeric GiNaC::mod (const numeric &a, const numeric &b)

  Modulus (in positive representation).
- const numeric GiNaC::smod (const numeric &a_, const numeric &b_)

  Modulus (in symmetric representation).
- const numeric GiNaC::irem (const numeric &a, const numeric &b)

  Numeric integer remainder.
- const numeric GiNaC::irem (const numeric &a, const numeric &b, numeric &q)

  Numeric integer remainder.
- const numeric GiNaC::iquo (const numeric &a, const numeric &b)

  Numeric integer quotient.
- const numeric GiNaC::iquo (const numeric &a, const numeric &b, numeric &r)

  Numeric integer quotient.
- const numeric GiNaC::gcd (const numeric &a, const numeric &b)

  Greatest Common Divisor.
- const numeric GiNaC::lcm (const numeric &a, const numeric &b)

  Least Common Multiple.
- const numeric GiNaC::pow (const numeric &x, const numeric &y)
- const numeric GiNaC::inverse (const numeric &x)
- numeric GiNaC::step (const numeric &x)
- int GiNaC::csgn (const numeric &x)
- bool GiNaC::is_zero (const numeric &x)
- bool GiNaC::is_positive (const numeric &x)
- bool GiNaC::is_negative (const numeric &x)
- bool GiNaC::is_integer (const numeric &x)
- bool GiNaC::is_pos_integer (const numeric &x)
- bool GiNaC::is_nonneg_integer (const numeric &x)
- bool GiNaC::is_even (const numeric &x)
- bool GiNaC::is_odd (const numeric &x)
- bool GiNaC::is_prime (const numeric &x)
- bool GiNaC::is_rational (const numeric &x)
- bool GiNaC::is_real (const numeric &x)
- bool GiNaC::is_cinteger (const numeric &x)
- bool GiNaC::is_crational (const numeric &x)
- int GiNaC::to_int (const numeric &x)
- long GiNaC::to_long (const numeric &x)
- double GiNaC::to_double (const numeric &x)
- const numeric GiNaC::real (const numeric &x)
- const numeric GiNaC::imag (const numeric &x)
- const numeric GiNaC::numer (const numeric &x)
- const numeric GiNaC::denom (const numeric &x)
- ex GiNaC::PiEvalf ()

  Floating point evaluation of Archimedes' constant Pi.
- ex GiNaC::EulerEvalf ()

  Floating point evaluation of Euler's constant gamma.
- ex GiNaC::CatalanEvalf ()

  Floating point evaluation of Catalan's constant.

9.69.1 Detailed Description

Makes the interface to the underlying bignum package available.

9.70 operators.cpp File Reference

Implementation of GiNaC's overloaded operators.

```
#include "operators.h"
#include "numeric.h"
#include "add.h"
#include "mul.h"
#include "power.h"
#include "ncmul.h"
#include "relational.h"
#include "print.h"
#include "utils.h"
```

Namespaces

- namespace GiNaC

Enumerations

- enum { GiNaC::callback_registered = 1 }

Functions

- static const ex GiNaC::exadd (const ex &lh, const ex &rh)

  Used internally by operator+() to add two ex objects.
- static const ex GiNaC::exmul (const ex &lh, const ex &rh)

  Used internally by operator∗() to multiply two ex objects.
- static const ex GiNaC::exminus (const ex &lh)

  Used internally by operator-() and friends to change the sign of an argument.
- const ex GiNaC::operator+ (const ex &lh, const ex &rh)
- const ex GiNaC::operator- (const ex &lh, const ex &rh)
- const ex GiNaC::operator∗ (const ex &lh, const ex &rh)
- const ex GiNaC::operator/ (const ex &lh, const ex &rh)
- const numeric GiNaC::operator+ (const numeric &lh, const numeric &rh)
- const numeric GiNaC::operator- (const numeric &lh, const numeric &rh)
- const numeric GiNaC::operator∗ (const numeric &lh, const numeric &rh)
- const numeric GiNaC::operator/ (const numeric &lh, const numeric &rh)
- ex & GiNaC::operator+= (ex &lh, const ex &rh)
- ex & GiNaC::operator-= (ex &lh, const ex &rh)
- ex & GiNaC::operator∗= (ex &lh, const ex &rh)
- ex & GiNaC::operator/= (ex &lh, const ex &rh)
- numeric & GiNaC::operator+= (numeric &lh, const numeric &rh)
- numeric & GiNaC::operator-= (numeric &lh, const numeric &rh)
- numeric & GiNaC::operator∗= (numeric &lh, const numeric &rh)

- numeric & GiNaC::operator/= (numeric &lh, const numeric &rh)
- const ex GiNaC::operator+ (const ex &lh)
- const ex GiNaC::operator- (const ex &lh)
- const numeric GiNaC::operator+ (const numeric &lh)
- const numeric GiNaC::operator- (const numeric &lh)
- ex & GiNaC::operator++ (ex &rh)

  Expression prefix increment.
- ex & GiNaC::operator-- (ex &rh)

  Expression prefix decrement.
- const ex GiNaC::operator++ (ex &lh, int)

  Expression postfix increment.
- const ex GiNaC::operator-- (ex &lh, int)

  Expression postfix decrement.
- numeric & GiNaC::operator++ (numeric &rh)

  Numeric prefix increment.
- numeric & GiNaC::operator-- (numeric &rh)

  Numeric prefix decrement.
- const numeric GiNaC::operator++ (numeric &lh, int)

  Numeric postfix increment.
- const numeric GiNaC::operator-- (numeric &lh, int)

  Numeric postfix decrement.
- const relational GiNaC::operator== (const ex &lh, const ex &rh)
- const relational GiNaC::operator!= (const ex &lh, const ex &rh)
- const relational GiNaC::operator< (const ex &lh, const ex &rh)
- const relational GiNaC::operator<= (const ex &lh, const ex &rh)
- const relational GiNaC::operator> (const ex &lh, const ex &rh)
- const relational GiNaC::operator>= (const ex &lh, const ex &rh)
- static int GiNaC::my_ios_index ()
- static void GiNaC::my_ios_callback (std::ios_base::event ev, std::ios_base &s, int i)
- static print_context * GiNaC::get_print_context (std::ios_base &s)
- static void GiNaC::set_print_context (std::ios_base &s, const print_context &c)
- static unsigned GiNaC::get_print_options (std::ios_base &s)
- static void GiNaC::set_print_options (std::ostream &s, unsigned options)
- std::ostream & GiNaC::operator<< (std::ostream &os, const ex &e)
- std::ostream & GiNaC::operator<< (std::ostream &os, const exvector &e)
- std::ostream & GiNaC::operator<< (std::ostream &os, const exset &e)
- std::ostream & GiNaC::operator<< (std::ostream &os, const exmap &e)
- std::istream & GiNaC::operator>> (std::istream &is, ex &e)
- std::ostream & GiNaC::dflt (std::ostream &os)
- std::ostream & GiNaC::latex (std::ostream &os)
- std::ostream & GiNaC::python (std::ostream &os)
- std::ostream & GiNaC::python_repr (std::ostream &os)
- std::ostream & GiNaC::tree (std::ostream &os)
- std::ostream & GiNaC::csrc (std::ostream &os)
- std::ostream & GiNaC::csrc_float (std::ostream &os)
- std::ostream & GiNaC::csrc_double (std::ostream &os)
- std::ostream & GiNaC::csrc_cl_N (std::ostream &os)
- std::ostream & GiNaC::index_dimensions (std::ostream &os)
- std::ostream & GiNaC::no_index_dimensions (std::ostream &os)

9.70.1 Detailed Description

Implementation of GiNaC's overloaded operators.

9.71 operators.h File Reference

Interface to GiNaC's overloaded operators.

```
#include <iosfwd>
```

Namespaces

- namespace GiNaC

Functions

- const ex GiNaC::operator+ (const ex &lh, const ex &rh)
- const ex GiNaC::operator- (const ex &lh, const ex &rh)
- const ex GiNaC::operator∗ (const ex &lh, const ex &rh)
- const ex GiNaC::operator/ (const ex &lh, const ex &rh)
- const numeric GiNaC::operator+ (const numeric &lh, const numeric &rh)
- const numeric GiNaC::operator- (const numeric &lh, const numeric &rh)
- const numeric GiNaC::operator∗ (const numeric &lh, const numeric &rh)
- const numeric GiNaC::operator/ (const numeric &lh, const numeric &rh)
- ex & GiNaC::operator+= (ex &lh, const ex &rh)
- ex & GiNaC::operator-= (ex &lh, const ex &rh)
- ex & GiNaC::operator∗= (ex &lh, const ex &rh)
- ex & GiNaC::operator/= (ex &lh, const ex &rh)
- numeric & GiNaC::operator+= (numeric &lh, const numeric &rh)
- numeric & GiNaC::operator-= (numeric &lh, const numeric &rh)
- numeric & GiNaC::operator∗= (numeric &lh, const numeric &rh)
- numeric & GiNaC::operator/= (numeric &lh, const numeric &rh)
- const ex GiNaC::operator+ (const ex &lh)
- const ex GiNaC::operator- (const ex &lh)
- const numeric GiNaC::operator+ (const numeric &lh)
- const numeric GiNaC::operator- (const numeric &lh)
- ex & GiNaC::operator++ (ex &rh)

  Expression prefix increment.
- ex & GiNaC::operator-- (ex &rh)

  Expression prefix decrement.
- const ex GiNaC::operator++ (ex &lh, int)

  Expression postfix increment.
- const ex GiNaC::operator-- (ex &lh, int)

  Expression postfix decrement.
- numeric & GiNaC::operator++ (numeric &rh)

  Numeric prefix increment.
- numeric & GiNaC::operator-- (numeric &rh)

  Numeric prefix decrement.
- const numeric GiNaC::operator++ (numeric &lh, int)

  Numeric postfix increment.
- const numeric GiNaC::operator-- (numeric &lh, int)

  Numeric postfix decrement.
- const relational GiNaC::operator== (const ex &lh, const ex &rh)
- const relational GiNaC::operator!= (const ex &lh, const ex &rh)

- const relational GiNaC::operator< (const ex &lh, const ex &rh)
- const relational GiNaC::operator<= (const ex &lh, const ex &rh)
- const relational GiNaC::operator> (const ex &lh, const ex &rh)
- const relational GiNaC::operator>= (const ex &lh, const ex &rh)
- std::ostream & GiNaC::operator<< (std::ostream &os, const ex &e)
- std::istream & GiNaC::operator>> (std::istream &is, ex &e)
- std::ostream & GiNaC::dflt (std::ostream &os)
- std::ostream & GiNaC::latex (std::ostream &os)
- std::ostream & GiNaC::python (std::ostream &os)
- std::ostream & GiNaC::python_repr (std::ostream &os)
- std::ostream & GiNaC::tree (std::ostream &os)
- std::ostream & GiNaC::csrc (std::ostream &os)
- std::ostream & GiNaC::csrc_float (std::ostream &os)
- std::ostream & GiNaC::csrc_double (std::ostream &os)
- std::ostream & GiNaC::csrc_cl_N (std::ostream &os)
- std::ostream & GiNaC::index_dimensions (std::ostream &os)
- std::ostream & GiNaC::no_index_dimensions (std::ostream &os)

9.71.1 Detailed Description

Interface to GiNaC's overloaded operators.

9.72 power.cpp File Reference

Implementation of GiNaC's symbolic exponentiation (basis^exponent).

```
#include "power.h"
#include "expairseq.h"
#include "add.h"
#include "mul.h"
#include "ncmul.h"
#include "numeric.h"
#include "constant.h"
#include "operators.h"
#include "inifcns.h"
#include "matrix.h"
#include "indexed.h"
#include "symbol.h"
#include "lst.h"
#include "archive.h"
#include "utils.h"
#include "relational.h"
#include "compiler.h"
#include <limits>
#include <stdexcept>
#include <vector>
#include <algorithm>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (power, basic, print_func< print_dflt >(&power::do_print_dflt). print_func< print_latex >(&power::do_print_latex). print_func< print_csrc >(&power::do_print_csrc). print_func< print_python >(&power::do_print_python). print_func< print_python_repr >(&power::do_print_python_repr). print_func< print_csrc_cl_N >(&power::do_print_csrc_cl_N)) power
- static void GiNaC::print_sym_pow (const print_context &c, const symbol &x, int exp)
- bool GiNaC::tryfactsubs (const ex &origfactor, const ex &patternfactor, int &nummatches, exmap &repls)
- GiNaC::GINAC_BIND_UNARCHIVER (power)

9.72.1 Detailed Description

Implementation of GiNaC's symbolic exponentiation (basis^exponent).

9.73 power.h File Reference

Interface to GiNaC's symbolic exponentiation (basis^exponent).

```
#include "basic.h"
#include "ex.h"
#include "archive.h"
```

Classes

- class GiNaC::power

  This class holds a two-component object, a basis and and exponent representing exponentiation.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (power)
- ex GiNaC::pow (const ex &b, const ex &e)

  Symbolic exponentiation.
- template<typename T1, typename T2>
  ex GiNaC::pow (const T1 &b, const T2 &e)
- ex GiNaC::sqrt (const ex &a)

  Square root expression.

9.73.1 Detailed Description

Interface to GiNaC's symbolic exponentiation (basis^exponent).

9.74 print.cpp File Reference

Implementation of helper classes for expression output.

```
#include "print.h"
#include <iostream>
```

Namespaces

- namespace GiNaC

Variables

- unsigned GiNaC::next_print_context_id = 0

  Next free ID for print_context types.

9.74.1 Detailed Description

Implementation of helper classes for expression output.

9.75 print.h File Reference

Definition of helper classes for expression output.

```
#include "class_info.h"
#include <iosfwd>
#include <memory>
```

Classes

- class GiNaC::print_context_options

  This class stores information about a registered print_context class.
- class GiNaC::print_options

  Flags to control the behavior of a print_context.
- class GiNaC::print_context

  Base class for print_contexts.
- class GiNaC::print_dflt

  Context for default (ginsh-parsable) output.
- class GiNaC::print_latex

  Context for latex-parsable output.
- class GiNaC::print_python

  Context for python pretty-print output.
- class GiNaC::print_python_repr

  Context for python-parsable output.
- class GiNaC::print_tree

Context for tree-like output for debugging.

- class GiNaC::print_csrc

  Base context for C source output.
- class GiNaC::print_csrc_float

  Context for C source output using float precision.
- class GiNaC::print_csrc_double

  Context for C source output using double precision.
- class GiNaC::print_csrc_cl_N

  Context for C source output using CLN numbers.
- class GiNaC::print_functor_impl

  Base class for print_functor handlers.
- class GiNaC::print_ptrfun_handler< T, C >

  print_functor handler for pointer-to-functions of class T, context type C
- class GiNaC::print_memfun_handler< T, C >

  print_functor handler for member functions of class T, context type C
- class GiNaC::print_functor

  This class represents a print method for a certain algebraic class and print_context type.

Namespaces

- namespace GiNaC

Macros

- #define GINAC_DECLARE_PRINT_CONTEXT_COMMON(classname)

  Common part of GINAC_DECLARE_PRINT_CONTEXT_BASE and GINAC_DECLARE_PRINT_CONTEXT_↩ DERIVED.
- #define GINAC_DECLARE_PRINT_CONTEXT_BASE(classname)
- #define GINAC_DECLARE_PRINT_CONTEXT(classname, supername)

  Macro for inclusion in the declaration of a print_context class.
- #define GINAC_IMPLEMENT_PRINT_CONTEXT(classname, supername)

  Macro for inclusion in the implementation of each print_context class.

Typedefs

- typedef class_info< print_context_options > GiNaC::print_context_class_info

Functions

- template<class T>
  bool GiNaC::is_a (const print_context &obj)

  Check if obj is a T, including base classes.

9.75.1 Detailed Description

Definition of helper classes for expression output.

9.75.2 Macro Definition Documentation

9.75.2.1 GINAC_DECLARE_PRINT_CONTEXT_COMMON

```
#define GINAC_DECLARE_PRINT_CONTEXT_COMMON(
                classname)
```

Value:

```
public: \
    friend class function_options; \
    friend class registered_class_options; \
    static const GiNaC::print_context_class_info &get_class_info_static(); \
    classname();
```

Common part of GINAC_DECLARE_PRINT_CONTEXT_BASE and GINAC_DECLARE_PRINT_CONTEXT_↩ DERIVED.

9.75.2.2 GINAC_DECLARE_PRINT_CONTEXT_BASE

```
#define GINAC_DECLARE_PRINT_CONTEXT_BASE(
                classname)
```

Value:

```
    GINAC_DECLARE_PRINT_CONTEXT_COMMON(classname) \
    virtual const GiNaC::print_context_class_info &get_class_info() const { return
      classname::get_class_info_static(); } \
    virtual const char *class_name() const { return classname::get_class_info_static().options.get_name(); }
      \
    virtual classname * duplicate() const { return new classname(*this); } \
private:
```

9.75.2.3 GINAC_DECLARE_PRINT_CONTEXT

```
#define GINAC_DECLARE_PRINT_CONTEXT(
                classname,
                supername)
```

Value:

```
    GINAC_DECLARE_PRINT_CONTEXT_COMMON(classname) \
    typedef supername inherited; \
    const GiNaC::print_context_class_info &get_class_info() const override { return
      classname::get_class_info_static(); } \
    const char *class_name() const override { return classname::get_class_info_static().options.get_name();
      } \
    classname * duplicate() const override { return new classname(*this); } \
private:
```

Macro for inclusion in the declaration of a print_context class.

It declares some functions that are common to all classes derived from 'print_context' as well as all required stuff for the GiNaC registry.

9.75.2.4 GINAC_IMPLEMENT_PRINT_CONTEXT

```
#define GINAC_IMPLEMENT_PRINT_CONTEXT(
                classname,
                supername)
```

Value:

```
const GiNaC::print_context_class_info &classname::get_class_info_static() \
{ \
    static GiNaC::print_context_class_info reg_info =
      GiNaC::print_context_class_info(GiNaC::print_context_options(#classname, #supername,
      GiNaC::next_print_context_id++)); \
    return reg_info; \
}
```

Macro for inclusion in the implementation of each print_context class.

9.76 pseries.cpp File Reference

Implementation of class for extended truncated power series and methods for series expansion.

```
#include "pseries.h"
#include "add.h"
#include "inifcns.h"
#include "lst.h"
#include "mul.h"
#include "power.h"
#include "relational.h"
#include "operators.h"
#include "symbol.h"
#include "integral.h"
#include "archive.h"
#include "utils.h"
#include <limits>
#include <numeric>
#include <stdexcept>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (pseries, basic, print_func< print_context >(&pseries::do_print). print_func< print_latex >(&pseries::do_print_latex). print_func< print_tree >(&pseries::do_print_tree). print_func< print_python >(&pseries::do_print_python). print_func< print_python_repr >(&pseries::do_print_python_repr)) pseries
- GiNaC::GINAC_BIND_UNARCHIVER (pseries)

9.76.1 Detailed Description

Implementation of class for extended truncated power series and methods for series expansion.

9.77 pseries.h File Reference

Interface to class for extended truncated power series.

```
#include "basic.h"
#include "expairseq.h"
```

Classes

- class GiNaC::pseries

  This class holds a extended truncated power series (positive and negative integer powers).

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (pseries)
- ex GiNaC::series_to_poly (const ex &e)

  Convert the pseries object embedded in an expression to an ordinary polynomial in the expansion variable.
- bool GiNaC::is_terminating (const pseries &s)

9.77.1 Detailed Description

Interface to class for extended truncated power series.

9.78 ptr.h File Reference

Reference-counted pointer template.

```
#include "assertion.h"
#include <cstddef>
#include <functional>
#include <iosfwd>
```

Classes

- class GiNaC::refcounted

  Base class for reference-counted objects.
- class GiNaC::ptr< T >

  Class of (intrusively) reference-counted pointers that support copy-on-write semantics.
- struct std::less< GiNaC::ptr< T > >

  Specialization of std::less for ptr<T> to enable ordering of ptr<T> objects (e.g.

Namespaces

- namespace GiNaC
- namespace std

9.78.1 Detailed Description

Reference-counted pointer template.

9.79 registrar.cpp File Reference

GiNaC's class registrar (for class basic and all classes derived from it).

```
#include "registrar.h"
```

Namespaces

- namespace GiNaC

9.79.1 Detailed Description

GiNaC's class registrar (for class basic and all classes derived from it).

9.80 registrar.h File Reference

GiNaC's class registrar (for class basic and all classes derived from it).

```
#include "class_info.h"
#include "print.h"
#include <list>
#include <typeinfo>
#include <vector>
```

Classes

- struct GiNaC::return_type_t

  To distinguish between different kinds of non-commutative objects.
- class GiNaC::registered_class_options

  This class stores information about a registered GiNaC class.

Namespaces

- namespace GiNaC

Macros

- #define GINAC_DECLARE_REGISTERED_CLASS_COMMON(classname)

  Common part of GINAC_DECLARE_REGISTERED_CLASS_NO_CTORS and GINAC_DECLARE_REGISTERED↩ _CLASS.
- #define GINAC_DECLARE_REGISTERED_CLASS_NO_CTORS(classname, supername)

  Primary macro for inclusion in the declaration of each registered class.
- #define GINAC_DECLARE_REGISTERED_CLASS(classname, supername)

  Macro for inclusion in the declaration of each registered class.
- #define GINAC_IMPLEMENT_REGISTERED_CLASS(classname, supername)

  Macro for inclusion in the implementation of each registered class.
- #define GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(classname, supername, options)

  Macro for inclusion in the implementation of each registered class.
- #define GINAC_IMPLEMENT_REGISTERED_CLASS_OPT_T(classname, supername, options)

  Macro for inclusion in the implementation of each registered class.

Typedefs

- typedef class_info< registered_class_options > GiNaC::registered_class_info

Functions

- template<typename T>
  return_type_t GiNaC::make_return_type_t (const unsigned rl=0)
- template<class Alg, class Ctx, class T, class C>
  void GiNaC::set_print_func (void f(const T &, const C &c, unsigned))

  Add or replace a print method.
- template<class Alg, class Ctx, class T, class C>
  void GiNaC::set_print_func (void(T::*f)(const C &, unsigned))

  Add or replace a print method.

9.80.1 Detailed Description

GiNaC's class registrar (for class basic and all classes derived from it).

9.80.2 Macro Definition Documentation

9.80.2.1 GINAC_DECLARE_REGISTERED_CLASS_COMMON

```
#define GINAC_DECLARE_REGISTERED_CLASS_COMMON(
              classname)
```

Value:

```
private: \
    static GiNaC::registered_class_info reg_info; \
public: \
    static GiNaC::registered_class_info &get_class_info_static() { return reg_info; } \
    class visitor { \
    public: \
        virtual void visit(const classname &) = 0; \
        virtual ~visitor() {}; \
    };
```

Common part of GINAC_DECLARE_REGISTERED_CLASS_NO_CTORS and GINAC_DECLARE_REGISTERED↩ _CLASS.

9.80.2.2 GINAC_DECLARE_REGISTERED_CLASS_NO_CTORS

```
#define GINAC_DECLARE_REGISTERED_CLASS_NO_CTORS(
                classname,
                supername)
```

Value:

```
    GINAC_DECLARE_REGISTERED_CLASS_COMMON(classname) \
    typedef supername inherited; \
    virtual const GiNaC::registered_class_info &get_class_info() const { return
      classname::get_class_info_static(); } \
    virtual GiNaC::registered_class_info &get_class_info() { return classname::get_class_info_static(); } \
    virtual const char *class_name() const { return classname::get_class_info_static().options.get_name(); }
      \
private:
```

Primary macro for inclusion in the declaration of each registered class.

9.80.2.3 GINAC_DECLARE_REGISTERED_CLASS

```
#define GINAC_DECLARE_REGISTERED_CLASS(
                classname,
                supername)
```

Value:

```
    GINAC_DECLARE_REGISTERED_CLASS_COMMON(classname) \
    template<class B, typename... Args> friend B & dynallocate(Args &&... args); \
    typedef supername inherited; \
    classname(); \
    classname * duplicate() const override { \
        classname * bp = new classname(*this); \
        bp->setflag(GiNaC::status_flags::dynallocated); \
        return bp; \
    } \
    \
    void accept(GiNaC::visitor & v) const override \
    { \
        if (visitor *p = dynamic_cast<visitor *>(&v)) \
            p->visit(*this); \
        else \
            inherited::accept(v); \
    } \
    const GiNaC::registered_class_info &get_class_info() const override { return
      classname::get_class_info_static(); } \
    GiNaC::registered_class_info &get_class_info() override { return classname::get_class_info_static(); } \
    const char *class_name() const override { return classname::get_class_info_static().options.get_name();
      } \
protected: \
    int compare_same_type(const GiNaC::basic & other) const override; \
private:
```

Macro for inclusion in the declaration of each registered class.

It declares some functions that are common to all classes derived from 'basic' as well as all required stuff for the GiNaC class registry (mainly needed for archiving).

Referenced by GiNaC::symmetry::sy_swap.

9.80.2.4 GINAC_IMPLEMENT_REGISTERED_CLASS

```
#define GINAC_IMPLEMENT_REGISTERED_CLASS(
                classname,
                supername)
```

Value:

```
GiNaC::registered_class_info classname::reg_info =
      GiNaC::registered_class_info(GiNaC::registered_class_options(#classname, #supername,
      typeid(classname)));
```

Macro for inclusion in the implementation of each registered class.

9.80.2.5 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT

```
#define GINAC_IMPLEMENT_REGISTERED_CLASS_OPT(
              classname,
              supername,
              options)
```

Value:

```
GiNaC::registered_class_info classname::reg_info =
      GiNaC::registered_class_info(GiNaC::registered_class_options(#classname, #supername,
      typeid(classname)).options);
```

Macro for inclusion in the implementation of each registered class.

Additional options can be specified.

9.80.2.6 GINAC_IMPLEMENT_REGISTERED_CLASS_OPT_T

```
#define GINAC_IMPLEMENT_REGISTERED_CLASS_OPT_T(
              classname,
              supername,
              options)
```

Value:

```
GiNaC::registered_class_info classname::reg_info =
      GiNaC::registered_class_info(GiNaC::registered_class_options(#classname, #supername,
      typeid(classname)).options);
```

Macro for inclusion in the implementation of each registered class.

Additional options can be specified.

9.81 relational.cpp File Reference

Implementation of relations between expressions.

```
#include "relational.h"
#include "operators.h"
#include "numeric.h"
#include "archive.h"
#include "utils.h"
#include "hash_seed.h"
#include <stdexcept>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (relational, basic, print_func< print_context >(&relational::do_print). print_func< print_tree >(&relational::do_print_tree). print_func< print_python_repr >(&relational::do_print_python_repr)) relational
- GiNaC::GINAC_BIND_UNARCHIVER (relational)
- static void GiNaC::print_operator (const print_context &c, relational::operators o)

9.81.1 Detailed Description

Implementation of relations between expressions.

9.82 relational.h File Reference

Interface to relations between expressions.

```
#include "basic.h"
#include "ex.h"
#include "archive.h"
```

Classes

- class GiNaC::relational

  This class holds a relation consisting of two expressions and a logical relation between them.
- struct GiNaC::relational::safe_bool_helper

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (relational)

9.82.1 Detailed Description

Interface to relations between expressions.

9.83 remember.cpp File Reference

Implementation of helper classes for using the remember option in GiNaC functions.

```
#include "function.h"
#include "utils.h"
#include "remember.h"
#include <stdexcept>
```

Namespaces

- namespace GiNaC

9.83.1 Detailed Description

Implementation of helper classes for using the remember option in GiNaC functions.

9.84 remember.h File Reference

Interface to helper classes for using the remember option in GiNaC functions.

```
#include <iosfwd>
#include <list>
#include <vector>
```

Classes

- class GiNaC::remember_table_entry

  A single entry in the remember table of a function.
- class GiNaC::remember_table_list

  A list of entries in the remember table having some least significant bits of the hashvalue in common.
- class GiNaC::remember_table

  The remember table is organized like an n-fold associative cache in a microprocessor.

Namespaces

- namespace GiNaC

9.84.1 Detailed Description

Interface to helper classes for using the remember option in GiNaC functions.

9.85 structure.h File Reference

Wrapper template for making GiNaC classes out of C++ structures.

```
#include "ex.h"
#include "ncmul.h"
#include "numeric.h"
#include "operators.h"
#include "print.h"
#include <functional>
```

Classes

- class GiNaC::compare_all_equal< T >

  Comparison policy: all structures of one type are equal.
- class GiNaC::compare_std_less< T >

  Comparison policy: use std::equal_to/std::less (defaults to operators == and <) to compare structures.
- class GiNaC::compare_bitwise< T >

  Comparison policy: use bit-wise comparison to compare structures.
- class GiNaC::structure< T, ComparisonPolicy >

  Wrapper template for making GiNaC classes out of C++ structures.

Namespaces

- namespace GiNaC

Variables

- template<class T, template< class > class CP>
  registered_class_info GiNaC::structure< T, CP >::reg_info = registered_class_info(registered_class_options(structure::get_cla
  "basic", typeid(structure<T, CP>)))

9.85.1 Detailed Description

Wrapper template for making GiNaC classes out of C++ structures.

9.86 symbol.cpp File Reference

Implementation of GiNaC's symbolic objects.

```
#include "symbol.h"
#include "lst.h"
#include "archive.h"
#include "utils.h"
#include "hash_seed.h"
#include "inifcns.h"
#include <map>
#include <stdexcept>
#include <string>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (symbol, basic, print_func< print_context >(&symbol::do_print). print_func< print_latex >(&symbol::do_print_latex). print_func< print_tree >(&symbol::do_print_tree). print_func< print_python_repr >(&symbol::do_print_python_repr)) symbol
- static const std::string & GiNaC::get_default_TeX_name (const std::string &name)

  Return default TeX name for symbol.
- GiNaC::GINAC_BIND_UNARCHIVER (symbol)
- GiNaC::GINAC_BIND_UNARCHIVER (realsymbol)
- GiNaC::GINAC_BIND_UNARCHIVER (possymbol)

9.86.1 Detailed Description

Implementation of GiNaC's symbolic objects.

9.87 symbol.h File Reference

Interface to GiNaC's symbolic objects.

```
#include "basic.h"
#include "ex.h"
#include "ptr.h"
#include "archive.h"
#include <string>
#include <typeinfo>
```

Classes

- class GiNaC::symbol

  Basic CAS symbol.
- class GiNaC::realsymbol

  Specialization of symbol to real domain.
- class GiNaC::possymbol

  Specialization of symbol to real positive domain.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (symbol)
- GiNaC::GINAC_DECLARE_UNARCHIVER (realsymbol)
- GiNaC::GINAC_DECLARE_UNARCHIVER (possymbol)

9.87.1 Detailed Description

Interface to GiNaC's symbolic objects.

9.88 symmetry.cpp File Reference

Implementation of GiNaC's symmetry definitions.

```
#include "symmetry.h"
#include "lst.h"
#include "add.h"
#include "numeric.h"
#include "operators.h"
#include "archive.h"
#include "utils.h"
#include "hash_seed.h"
#include <functional>
#include <limits>
#include <stdexcept>
```

Classes

- class GiNaC::sy_is_less
- class GiNaC::sy_swap

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (symmetry, basic, print_func< print_context >(&symmetry::do_print). print_func< print_tree >(&symmetry::do_print_tree)) symmetry
- GiNaC::GINAC_BIND_UNARCHIVER (symmetry)
- static const symmetry & GiNaC::index0 ()
- static const symmetry & GiNaC::index1 ()
- static const symmetry & GiNaC::index2 ()
- static const symmetry & GiNaC::index3 ()
- const symmetry & GiNaC::not_symmetric ()
- const symmetry & GiNaC::symmetric2 ()
- const symmetry & GiNaC::symmetric3 ()
- const symmetry & GiNaC::symmetric4 ()
- const symmetry & GiNaC::antisymmetric2 ()
- const symmetry & GiNaC::antisymmetric3 ()
- const symmetry & GiNaC::antisymmetric4 ()
- int GiNaC::canonicalize (exvector::iterator v, const symmetry &symm)

  Canonicalize the order of elements of an expression vector, according to the symmetry properties defined in a symmetry tree.
- static ex GiNaC::symm (const ex &e, exvector::const_iterator first, exvector::const_iterator last, bool asymmetric)
- ex GiNaC::symmetrize (const ex &e, exvector::const_iterator first, exvector::const_iterator last)

  Symmetrize expression over a set of objects (symbols, indices).
- ex GiNaC::antisymmetrize (const ex &e, exvector::const_iterator first, exvector::const_iterator last)

  Antisymmetrize expression over a set of objects (symbols, indices).
- ex GiNaC::symmetrize_cyclic (const ex &e, exvector::const_iterator first, exvector::const_iterator last)

  Symmetrize expression by cyclic permutation over a set of objects (symbols, indices).

9.88.1 Detailed Description

Implementation of GiNaC's symmetry definitions.

9.89 symmetry.h File Reference

Interface to GiNaC's symmetry definitions.

```
#include "ex.h"
#include "archive.h"
#include <set>
```

Classes

- class GiNaC::symmetry

  This class describes the symmetry of a group of indices.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (symmetry)
- symmetry GiNaC::sy_none ()
- symmetry GiNaC::sy_none (const symmetry &c1, const symmetry &c2)
- symmetry GiNaC::sy_none (const symmetry &c1, const symmetry &c2, const symmetry &c3)
- symmetry GiNaC::sy_none (const symmetry &c1, const symmetry &c2, const symmetry &c3, const symmetry &c4)
- symmetry GiNaC::sy_symm ()
- symmetry GiNaC::sy_symm (const symmetry &c1, const symmetry &c2)
- symmetry GiNaC::sy_symm (const symmetry &c1, const symmetry &c2, const symmetry &c3)
- symmetry GiNaC::sy_symm (const symmetry &c1, const symmetry &c2, const symmetry &c3, const symmetry &c4)
- symmetry GiNaC::sy_anti ()
- symmetry GiNaC::sy_anti (const symmetry &c1, const symmetry &c2)
- symmetry GiNaC::sy_anti (const symmetry &c1, const symmetry &c2, const symmetry &c3)
- symmetry GiNaC::sy_anti (const symmetry &c1, const symmetry &c2, const symmetry &c3, const symmetry &c4)
- symmetry GiNaC::sy_cycl ()
- symmetry GiNaC::sy_cycl (const symmetry &c1, const symmetry &c2)
- symmetry GiNaC::sy_cycl (const symmetry &c1, const symmetry &c2, const symmetry &c3)
- symmetry GiNaC::sy_cycl (const symmetry &c1, const symmetry &c2, const symmetry &c3, const symmetry &c4)
- const symmetry & GiNaC::not_symmetric ()
- const symmetry & GiNaC::symmetric2 ()
- const symmetry & GiNaC::symmetric3 ()
- const symmetry & GiNaC::symmetric4 ()
- const symmetry & GiNaC::antisymmetric2 ()

- const symmetry & GiNaC::antisymmetric3 ()
- const symmetry & GiNaC::antisymmetric4 ()
- int GiNaC::canonicalize (exvector::iterator v, const symmetry &symm)

  Canonicalize the order of elements of an expression vector, according to the symmetry properties defined in a symmetry tree.
- ex GiNaC::symmetrize (const ex &e, exvector::const_iterator first, exvector::const_iterator last)

  Symmetrize expression over a set of objects (symbols, indices).
- ex GiNaC::symmetrize (const ex &e, const exvector &v)

  Symmetrize expression over a set of objects (symbols, indices).
- ex GiNaC::antisymmetrize (const ex &e, exvector::const_iterator first, exvector::const_iterator last)

  Antisymmetrize expression over a set of objects (symbols, indices).
- ex GiNaC::antisymmetrize (const ex &e, const exvector &v)

  Antisymmetrize expression over a set of objects (symbols, indices).
- ex GiNaC::symmetrize_cyclic (const ex &e, exvector::const_iterator first, exvector::const_iterator last)

  Symmetrize expression by cyclic permutation over a set of objects (symbols, indices).
- ex GiNaC::symmetrize_cyclic (const ex &e, const exvector &v)

  Symmetrize expression by cyclic permutation over a set of objects (symbols, indices).

9.89.1 Detailed Description

Interface to GiNaC's symmetry definitions.

9.90 tensor.cpp File Reference

Implementation of GiNaC's special tensors.

```
#include "tensor.h"
#include "idx.h"
#include "indexed.h"
#include "symmetry.h"
#include "relational.h"
#include "operators.h"
#include "lst.h"
#include "numeric.h"
#include "matrix.h"
#include "archive.h"
#include "utils.h"
#include <stdexcept>
#include <vector>
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (tensdelta, tensor, print_func< print_dflt >(&tensdelta::do_print). print_func< print_latex >(&tensdelta::do_print_latex)) GINAC_IMPLEMENT_↩ REGISTERED_CLASS_OPT(tensmetric
- GiNaC::print_func< print_dflt > (&tensmetric::do_print). print_func< print_latex >(&tensmetric
- GiNaC::GINAC_BIND_UNARCHIVER (minkmetric)
- GiNaC::GINAC_BIND_UNARCHIVER (tensepsilon)
- GiNaC::GINAC_BIND_UNARCHIVER (tensdelta)
- GiNaC::GINAC_BIND_UNARCHIVER (tensmetric)
- GiNaC::GINAC_BIND_UNARCHIVER (spinmetric)
- ex GiNaC::delta_tensor (const ex &i1, const ex &i2)

  Create a delta tensor with specified indices.
- ex GiNaC::metric_tensor (const ex &i1, const ex &i2)

  Create a symmetric metric tensor with specified indices.
- ex GiNaC::lorentz_g (const ex &i1, const ex &i2, bool pos_sig=false)

  Create a Minkowski metric tensor with specified indices.
- ex GiNaC::spinor_metric (const ex &i1, const ex &i2)

  Create a spinor metric tensor with specified indices.
- ex GiNaC::epsilon_tensor (const ex &i1, const ex &i2)

  Create an epsilon tensor in a Euclidean space with two indices.
- ex GiNaC::epsilon_tensor (const ex &i1, const ex &i2, const ex &i3)

  Create an epsilon tensor in a Euclidean space with three indices.
- ex GiNaC::lorentz_eps (const ex &i1, const ex &i2, const ex &i3, const ex &i4, bool pos_sig=false)

  Create an epsilon tensor in a Minkowski space with four indices.

9.90.1 Detailed Description

Implementation of GiNaC's special tensors.

9.91 tensor.h File Reference

Interface to GiNaC's special tensors.

```
#include "ex.h"
#include "archive.h"
```

Classes

- class GiNaC::tensor

  This class holds one of GiNaC's predefined special tensors such as the delta and the metric tensors.
- class GiNaC::tensdelta

  This class represents the delta tensor.
- class GiNaC::tensmetric

  This class represents a general metric tensor which can be used to raise/lower indices.
- class GiNaC::minkmetric

  This class represents a Minkowski metric tensor.
- class GiNaC::spinmetric

  This class represents an antisymmetric spinor metric tensor which can be used to raise/lower indices of 2-component Weyl spinors.
- class GiNaC::tensepsilon

  This class represents the totally antisymmetric epsilon tensor.

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (tensdelta)
- GiNaC::GINAC_DECLARE_UNARCHIVER (tensmetric)
- GiNaC::GINAC_DECLARE_UNARCHIVER (minkmetric)
- GiNaC::GINAC_DECLARE_UNARCHIVER (spinmetric)
- GiNaC::GINAC_DECLARE_UNARCHIVER (tensepsilon)
- ex GiNaC::delta_tensor (const ex &i1, const ex &i2)

  Create a delta tensor with specified indices.
- ex GiNaC::metric_tensor (const ex &i1, const ex &i2)

  Create a symmetric metric tensor with specified indices.
- ex GiNaC::lorentz_g (const ex &i1, const ex &i2, bool pos_sig=false)

  Create a Minkowski metric tensor with specified indices.
- ex GiNaC::spinor_metric (const ex &i1, const ex &i2)

  Create a spinor metric tensor with specified indices.
- ex GiNaC::epsilon_tensor (const ex &i1, const ex &i2)

  Create an epsilon tensor in a Euclidean space with two indices.
- ex GiNaC::epsilon_tensor (const ex &i1, const ex &i2, const ex &i3)

  Create an epsilon tensor in a Euclidean space with three indices.
- ex GiNaC::lorentz_eps (const ex &i1, const ex &i2, const ex &i3, const ex &i4, bool pos_sig=false)

  Create an epsilon tensor in a Minkowski space with four indices.

9.91.1 Detailed Description

Interface to GiNaC's special tensors.

9.92 utils.cpp File Reference

Implementation of several small and furry utilities needed within GiNaC but not of any interest to the user of the library.

```
#include "ex.h"
#include "numeric.h"
#include "utils.h"
#include "version.h"
```

Namespaces

- namespace GiNaC

Functions

- unsigned GiNaC::log2 (unsigned n)

  Integer binary logarithm.
- const numeric GiNaC::multinomial_coefficient (const std::vector< unsigned > &p)

  Compute the multinomial coefficient n!

Variables

- const int GiNaC::version_major = GINACLIB_MAJOR_VERSION
- const int GiNaC::version_minor = GINACLIB_MINOR_VERSION
- const int GiNaC::version_micro = GINACLIB_MICRO_VERSION
- const numeric * GiNaC::_num_120_p
- const ex GiNaC::_ex_120 = ex(*_num_120_p)
- const numeric * GiNaC::_num_60_p
- const ex GiNaC::_ex_60 = ex(*_num_60_p)
- const numeric * GiNaC::_num_48_p
- const ex GiNaC::_ex_48 = ex(*_num_48_p)
- const numeric * GiNaC::_num_30_p
- const ex GiNaC::_ex_30 = ex(*_num_30_p)
- const numeric * GiNaC::_num_25_p
- const ex GiNaC::_ex_25 = ex(*_num_25_p)
- const numeric * GiNaC::_num_24_p
- const ex GiNaC::_ex_24 = ex(*_num_24_p)
- const numeric * GiNaC::_num_20_p
- const ex GiNaC::_ex_20 = ex(*_num_20_p)
- const numeric * GiNaC::_num_18_p
- const ex GiNaC::_ex_18 = ex(*_num_18_p)
- const numeric * GiNaC::_num_15_p
- const ex GiNaC::_ex_15 = ex(*_num_15_p)
- const numeric * GiNaC::_num_12_p
- const ex GiNaC::_ex_12 = ex(*_num_12_p)
- const numeric * GiNaC::_num_11_p
- const ex GiNaC::_ex_11 = ex(*_num_11_p)
- const numeric * GiNaC::_num_10_p
- const ex GiNaC::_ex_10 = ex(*_num_10_p)
- const numeric * GiNaC::_num_9_p
- const ex GiNaC::_ex_9 = ex(*_num_9_p)
- const numeric * GiNaC::_num_8_p
- const ex GiNaC::_ex_8 = ex(*_num_8_p)
- const numeric * GiNaC::_num_7_p
- const ex GiNaC::_ex_7 = ex(*_num_7_p)
- const numeric * GiNaC::_num_6_p
- const ex GiNaC::_ex_6 = ex(*_num_6_p)
- const numeric * GiNaC::_num_5_p
- const ex GiNaC::_ex_5 = ex(*_num_5_p)
- const numeric * GiNaC::_num_4_p
- const ex GiNaC::_ex_4 = ex(*_num_4_p)
- const numeric * GiNaC::_num_3_p
- const ex GiNaC::_ex_3 = ex(*_num_3_p)
- const numeric * GiNaC::_num_2_p
- const ex GiNaC::_ex_2 = ex(*_num_2_p)
- const numeric * GiNaC::_num_1_p
- const ex GiNaC::_ex_1 = ex(*_num_1_p)

- const numeric * GiNaC::_num_1_2_p
- const ex GiNaC::_ex_1_2 = ex(*_num_1_2_p)
- const numeric * GiNaC::_num_1_3_p
- const ex GiNaC::_ex_1_3 = ex(*_num_1_3_p)
- const numeric * GiNaC::_num_1_4_p
- const ex GiNaC::_ex_1_4 = ex(*_num_1_4_p)
- const numeric * GiNaC::_num0_p
- const ex GiNaC::_ex0 = ex(*_num0_p)
- const numeric * GiNaC::_num1_4_p
- const ex GiNaC::_ex1_4 = ex(*_num1_4_p)
- const numeric * GiNaC::_num1_3_p
- const ex GiNaC::_ex1_3 = ex(*_num1_3_p)
- const numeric * GiNaC::_num1_2_p
- const ex GiNaC::_ex1_2 = ex(*_num1_2_p)
- const numeric * GiNaC::_num1_p
- const ex GiNaC::_ex1 = ex(*_num1_p)
- const numeric * GiNaC::_num2_p
- const ex GiNaC::_ex2 = ex(*_num2_p)
- const numeric * GiNaC::_num3_p
- const ex GiNaC::_ex3 = ex(*_num3_p)
- const numeric * GiNaC::_num4_p
- const ex GiNaC::_ex4 = ex(*_num4_p)
- const numeric * GiNaC::_num5_p
- const ex GiNaC::_ex5 = ex(*_num5_p)
- const numeric * GiNaC::_num6_p
- const ex GiNaC::_ex6 = ex(*_num6_p)
- const numeric * GiNaC::_num7_p
- const ex GiNaC::_ex7 = ex(*_num7_p)
- const numeric * GiNaC::_num8_p
- const ex GiNaC::_ex8 = ex(*_num8_p)
- const numeric * GiNaC::_num9_p
- const ex GiNaC::_ex9 = ex(*_num9_p)
- const numeric * GiNaC::_num10_p
- const ex GiNaC::_ex10 = ex(*_num10_p)
- const numeric * GiNaC::_num11_p
- const ex GiNaC::_ex11 = ex(*_num11_p)
- const numeric * GiNaC::_num12_p
- const ex GiNaC::_ex12 = ex(*_num12_p)
- const numeric * GiNaC::_num15_p
- const ex GiNaC::_ex15 = ex(*_num15_p)
- const numeric * GiNaC::_num18_p
- const ex GiNaC::_ex18 = ex(*_num18_p)
- const numeric * GiNaC::_num20_p
- const ex GiNaC::_ex20 = ex(*_num20_p)
- const numeric * GiNaC::_num24_p
- const ex GiNaC::_ex24 = ex(*_num24_p)
- const numeric * GiNaC::_num25_p
- const ex GiNaC::_ex25 = ex(*_num25_p)
- const numeric * GiNaC::_num30_p
- const ex GiNaC::_ex30 = ex(*_num30_p)
- const numeric * GiNaC::_num48_p
- const ex GiNaC::_ex48 = ex(*_num48_p)
- const numeric * GiNaC::_num60_p
- const ex GiNaC::_ex60 = ex(*_num60_p)
- const numeric * GiNaC::_num120_p
- const ex GiNaC::_ex120 = ex(*_num120_p)

9.92.1 Detailed Description

Implementation of several small and furry utilities needed within GiNaC but not of any interest to the user of the library.

9.93 utils.h File Reference

Interface to several small and furry utilities needed within GiNaC but not of any interest to the user of the library.

```
#include "assertion.h"
#include <functional>
#include <cstdint>
#include <string>
```

Classes

- class GiNaC::dunno

  Exception class thrown by functions to signal unimplemented functionality so the expression may just be .hold().
- class GiNaC::basic_partition_generator

  Base class for generating all bounded combinatorial partitions of an integer n with exactly m parts in non-decreasing order.
- struct GiNaC::basic_partition_generator::mpartition2
- class GiNaC::partition_with_zero_parts_generator

  Generate all bounded combinatorial partitions of an integer n with exactly m parts (including zero parts) in non-decreasing order.
- class GiNaC::partition_generator

  Generate all bounded combinatorial partitions of an integer n with exactly m parts (not including zero parts) in non-decreasing order.
- class GiNaC::composition_generator

  Generate all compositions of a partition of an integer n, starting with the compositions which has non-decreasing order.
- struct GiNaC::composition_generator::coolmulti
- struct GiNaC::composition_generator::coolmulti::element

Namespaces

- namespace GiNaC

Macros

- #define DEFAULT_CTOR(classname)
- #define DEFAULT_COMPARE(classname)
- #define DEFAULT_PRINT(classname, text)
- #define DEFAULT_PRINT_LATEX(classname, text, latex)

Functions

- unsigned GiNaC::log2 (unsigned n)

  Integer binary logarithm.
- unsigned GiNaC::rotate_left (unsigned n)

  Rotate bits of unsigned value by one bit to the left.
- template<class T>
  int GiNaC::compare_pointers (const T *a, const T *b)

  Compare two pointers (just to establish some sort of canonical order).
- unsigned GiNaC::golden_ratio_hash (uintptr_t n)

  Truncated multiplication with golden ratio, for computing hash values.
- template<class It>
  int GiNaC::permutation_sign (It first, It last)
- template<class It, class Cmp, class Swap>
  int GiNaC::permutation_sign (It first, It last, Cmp comp, Swap swapit)
- template<class It, class Cmp, class Swap>
  void GiNaC::shaker_sort (It first, It last, Cmp comp, Swap swapit)
- template<class It, class Swap>
  void GiNaC::cyclic_permutation (It first, It last, It new_first, Swap swapit)
- const numeric GiNaC::multinomial_coefficient (const std::vector< unsigned > &p)

  Compute the multinomial coefficient n!

9.93.1 Detailed Description

Interface to several small and furry utilities needed within GiNaC but not of any interest to the user of the library.

9.93.2 Macro Definition Documentation

9.93.2.1 DEFAULT_CTOR

```
#define DEFAULT_CTOR(
              classname)
```

Value:

```
classname::classname() { setflag(status_flags::evaluated | status_flags::expanded); }
```

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

9.93.2.2 DEFAULT_COMPARE

```
#define DEFAULT_COMPARE(
              classname)
```

Value:

```
int classname::compare_same_type(const basic & other) const \
{ \
      \
    return 0; \
}
```

Referenced by GiNaC::GINAC_BIND_UNARCHIVER(), and GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

9.93.2.3 DEFAULT_PRINT

```
#define DEFAULT_PRINT(
                classname,
                text)
```

Value:

```
void classname::do_print(const print_context & c, unsigned level) const \
{ \
    c.s « text; \
}
```

Referenced by GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT().

9.93.2.4 DEFAULT_PRINT_LATEX

```
#define DEFAULT_PRINT_LATEX(
                classname,
                text,
                latex)
```

Value:

```
DEFAULT_PRINT(classname, text) \
void classname::do_print_latex(const print_latex & c, unsigned level) const \
{ \
    c.s « latex; \
}
```

9.94 utils_multi_iterator.h File Reference

Utilities for summing over multiple indices.

```
#include <cstddef>
#include <vector>
#include <ostream>
#include <iterator>
```

Classes

- class GiNaC::has_distance< T >
  SFINAE test for distance.
- class GiNaC::basic_multi_iterator< T >
  basic_multi_iterator is a base class.
- class GiNaC::multi_iterator_ordered< T >
  The class multi_iterator_ordered defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.
- class GiNaC::multi_iterator_ordered_eq< T >
  The class multi_iterator_ordered_eq defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.
- class GiNaC::multi_iterator_ordered_eq_indv< T >
  The class multi_iterator_ordered_eq_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.
- class GiNaC::multi_iterator_counter< T >
  The class multi_iterator_counter defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.
- class GiNaC::multi_iterator_counter_indv< T >

The class multi_iterator_counter_indv defines a multi_iterator $(i_1, i_2, ..., i_k)$, such that.

- class GiNaC::multi_iterator_permutation< T >

  The class multi_iterator_permutation defines a multi_iterator $(i_1, i_2, ..., i_k)$, for which.

- class GiNaC::multi_iterator_shuffle< T >

  The class multi_iterator_shuffle defines a multi_iterator, which runs over all shuffles of a and b.

- class GiNaC::multi_iterator_shuffle_prime< T >

  The class multi_iterator_shuffle_prime defines a multi_iterator, which runs over all shuffles of a and b, excluding the first one (a,b).

Namespaces

- namespace GiNaC

Functions

- template<typename T>
  std::enable_if< has_distance< T >::value, typenamestd::iterator_traits< T >::difference_type >::type GiNaC::format_index_value (const T &a, const T &b)

  For printing a multi-index: If the templates are used, where T is an iterator, printing the address where the iterator points to is not meaningful.

- template<typename T>
  std::enable_if<!has_distance< T >::value, T >::type GiNaC::format_index_value (const T &a, const T &b)

  For all other cases we simply print the value.

- template<class T>
  std::ostream & GiNaC::operator<< (std::ostream &os, const basic_multi_iterator< T > &v)

  Output operator.

- template<class T>
  std::ostream & GiNaC::operator<< (std::ostream &os, const multi_iterator_ordered< T > &v)

  Output operator.

- template<class T>
  std::ostream & GiNaC::operator<< (std::ostream &os, const multi_iterator_ordered_eq< T > &v)

  Output operator.

- template<class T>
  std::ostream & GiNaC::operator<< (std::ostream &os, const multi_iterator_ordered_eq_indv< T > &v)

  Output operator.

- template<class T>
  std::ostream & GiNaC::operator<< (std::ostream &os, const multi_iterator_counter< T > &v)

  Output operator.

- template<class T>
  std::ostream & GiNaC::operator<< (std::ostream &os, const multi_iterator_counter_indv< T > &v)

  Output operator.

- template<class T>
  std::ostream & GiNaC::operator<< (std::ostream &os, const multi_iterator_permutation< T > &v)

  Output operator.

- template<class T>
  std::ostream & GiNaC::operator<< (std::ostream &os, const multi_iterator_shuffle< T > &v)

  Output operator.

- template<class T>
  std::ostream & GiNaC::operator<< (std::ostream &os, const multi_iterator_shuffle_prime< T > &v)

  Output operator.

9.94.1 Detailed Description

Utilities for summing over multiple indices.

9.95 version.h File Reference

GiNaC library version information.

Namespaces

- namespace GiNaC

Macros

- #define GINACLIB_MAJOR_VERSION 1
- #define GINACLIB_MINOR_VERSION 8
- #define GINACLIB_MICRO_VERSION 10
- #define GINAC_LT_CURRENT 14
- #define GINAC_LT_REVISION 2
- #define GINAC_LT_AGE 1
- #define GINACLIB_ARCHIVE_VERSION 3
- #define GINACLIB_ARCHIVE_AGE 3
- #define GINACLIB_STR_HELPER(x)
- #define GINACLIB_STR(x)
- #define GINACLIB_VERSION

9.95.1 Detailed Description

GiNaC library version information.

9.95.2 Macro Definition Documentation

9.95.2.1 GINACLIB_MAJOR_VERSION

```
#define GINACLIB_MAJOR_VERSION 1
```

9.95.2.2 GINACLIB_MINOR_VERSION

```
#define GINACLIB_MINOR_VERSION 8
```

9.95.2.3 GINACLIB_MICRO_VERSION

```
#define GINACLIB_MICRO_VERSION 10
```

9.95.2.4 GINAC_LT_CURRENT

```
#define GINAC_LT_CURRENT 14
```

9.95.2.5 GINAC_LT_REVISION

```
#define GINAC_LT_REVISION 2
```

9.95.2.6 GINAC_LT_AGE

```
#define GINAC_LT_AGE 1
```

9.95.2.7 GINACLIB_ARCHIVE_VERSION

```
#define GINACLIB_ARCHIVE_VERSION 3
```

Referenced by GiNaC::archive::operator<<, and GiNaC::archive::operator>>.

9.95.2.8 GINACLIB_ARCHIVE_AGE

```
#define GINACLIB_ARCHIVE_AGE 3
```

Referenced by GiNaC::archive::operator>>.

9.95.2.9 GINACLIB_STR_HELPER

```
#define GINACLIB_STR_HELPER(
              x)
```

Value:

```
#x
```

9.95.2.10 GINACLIB_STR

```
#define GINACLIB_STR(
              x)
```

Value:

```
GINACLIB_STR_HELPER(x)
```

9.95.2.11 GINACLIB_VERSION

```
#define GINACLIB_VERSION
```

Value:

```
GINACLIB_STR(GINACLIB_MAJOR_VERSION) "." \
GINACLIB_STR(GINACLIB_MINOR_VERSION) "." \
GINACLIB_STR(GINACLIB_MICRO_VERSION)
```

9.96 wildcard.cpp File Reference

Implementation of GiNaC's wildcard objects.

```
#include "wildcard.h"
#include "archive.h"
#include "utils.h"
#include "hash_seed.h"
```

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_IMPLEMENT_REGISTERED_CLASS_OPT (wildcard, basic, print_func< print_context >(&wildcard::do_print). print_func< print_tree >(&wildcard::do_print_tree). print_func< print_python_repr >(&wildcard::do_print_python_repr)) wildcard
- GiNaC::GINAC_BIND_UNARCHIVER (wildcard)
- bool GiNaC::haswild (const ex &x)

  Check whether x has a wildcard anywhere as a subexpression.

9.96.1 Detailed Description

Implementation of GiNaC's wildcard objects.

9.97 wildcard.h File Reference

Interface to GiNaC's wildcard objects.

```
#include "ex.h"
#include "archive.h"
```

Classes

- class GiNaC::wildcard

  This class acts as a wildcard for subs(), match(), has() and find().

Namespaces

- namespace GiNaC

Functions

- GiNaC::GINAC_DECLARE_UNARCHIVER (wildcard)
- ex GiNaC::wild (unsigned label=0)

  Create a wildcard object with the specified label.
- bool GiNaC::haswild (const ex &x)

  Check whether x has a wildcard anywhere as a subexpression.

9.97.1 Detailed Description

Interface to GiNaC's wildcard objects.

Index

_ex0
GiNaC, 303
_ex1
GiNaC, 305
_ex10
GiNaC, 308
_ex11
GiNaC, 308
_ex12
GiNaC, 309
_ex120
GiNaC, 311
_ex15
GiNaC, 309
_ex18
GiNaC, 309
_ex1_2
GiNaC, 305
_ex1_3
GiNaC, 304
_ex1_4
GiNaC, 304
_ex2
GiNaC, 306
_ex20
GiNaC, 310
_ex24
GiNaC, 310
_ex25
GiNaC, 310
_ex3
GiNaC, 306
_ex30
GiNaC, 310
_ex4
GiNaC, 306
_ex48
GiNaC, 311
_ex5
GiNaC, 307
_ex6
GiNaC, 307
_ex60
GiNaC, 311
_ex7
GiNaC, 307
_ex8
GiNaC, 308
_ex9
GiNaC, 308
_ex_1
GiNaC, 302
_ex_10
GiNaC, 300
_ex_11
GiNaC, 299
_ex_12
GiNaC, 299
_ex_120
GiNaC, 297
_ex_15
GiNaC, 299
_ex_18
GiNaC, 299
_ex_1_2
GiNaC, 302
_ex_1_3
GiNaC, 303
_ex_1_4
GiNaC, 303
_ex_2
GiNaC, 302
_ex_20
GiNaC, 298
_ex_24
GiNaC, 298
_ex_25
GiNaC, 298
_ex_3
GiNaC, 301
_ex_30
GiNaC, 298
_ex_4
GiNaC, 301
_ex_48
GiNaC, 297
_ex_5
GiNaC, 301
_ex_6
GiNaC, 301
_ex_60
GiNaC, 297
_ex_7
GiNaC, 300
_ex_8
GiNaC, 300
_ex_9
GiNaC, 300

_iter_rep
GiNaC::internal::_iter_rep, 316
_num0_bp
GiNaC, 295
_num0_p
GiNaC, 303
_num10_p
GiNaC, 308
_num11_p
GiNaC, 308
_num120_p
GiNaC, 311
_num12_p
GiNaC, 309
_num15_p
GiNaC, 309
_num18_p
GiNaC, 309
_num1_2_p
GiNaC, 304
_num1_3_p
GiNaC, 304
_num1_4_p
GiNaC, 304
_num1_p
GiNaC, 305
_num20_p
GiNaC, 309
_num24_p
GiNaC, 310
_num25_p
GiNaC, 310
_num2_p
GiNaC, 306
_num30_p
GiNaC, 310
_num3_p
GiNaC, 306
_num48_p
GiNaC, 310
_num4_p
GiNaC, 306
_num5_p
GiNaC, 307
_num60_p
GiNaC, 311
_num6_p
GiNaC, 307
_num7_p
GiNaC, 307
_num8_p
GiNaC, 307
_num9_p
GiNaC, 308
_num_10_p
GiNaC, 299
_num_11_p
GiNaC, 299
_num_120_p
GiNaC, 297
_num_12_p
GiNaC, 299
_num_15_p
GiNaC, 299
_num_18_p
GiNaC, 298
_num_1_2_p
GiNaC, 302
_num_1_3_p
GiNaC, 302
_num_1_4_p
GiNaC, 303
_num_1_p
GiNaC, 302
_num_20_p
GiNaC, 298
_num_24_p
GiNaC, 298
_num_25_p
GiNaC, 298
_num_2_p
GiNaC, 301
_num_30_p
GiNaC, 297
_num_3_p
GiNaC, 301
_num_48_p
GiNaC, 297
_num_4_p
GiNaC, 301
_num_5_p
GiNaC, 301
_num_60_p
GiNaC, 297
_num_6_p
GiNaC, 300
_num_7_p
GiNaC, 300
_num_8_p
GiNaC, 300
_num_9_p
GiNaC, 300
_numeric_digits
GiNaC::_numeric_digits, 318
∼archive
GiNaC::archive, 338
∼basic
GiNaC::basic, 363
∼basic_multi_iterator
GiNaC::basic_multi_iterator< T >, 396
∼compare_all_equal
GiNaC::compare_all_equal< T >, 437
∼compare_bitwise
GiNaC::compare_bitwise< T >, 438
∼compare_std_less
GiNaC::compare_std_less< T >, 439

~container_storage
GiNaC::container_storage< C >, 483
~coolmulti
GiNaC::composition_generator::coolmulti, 485
~element
GiNaC::composition_generator::coolmulti::element, 546
~function_options
GiNaC::function_options, 671
~library_init
GiNaC::library_init, 808
~map_function
GiNaC::map_function, 813
~print_context
GiNaC::print_context, 1014
~print_functor_impl
GiNaC::print_functor_impl, 1029
~ptr
GiNaC::ptr< T >, 1066
~unarchive_table_t
GiNaC::unarchive_table_t, 1234
~visitor
GiNaC::visitor, 1252

a
GiNaC::archive_node, 354
GiNaC::Eisenstein_kernel, 545
GiNaC::integral, 772
abs
GiNaC, 248
abs_conjugate
GiNaC, 150
abs_eval
GiNaC, 149
abs_evalf
GiNaC, 149
abs_expand
GiNaC, 149
abs_expl_derivative
GiNaC, 150
abs_imag_part
GiNaC, 151
abs_info
GiNaC, 151
abs_power
GiNaC, 151
abs_print_csrc_float
GiNaC, 150
abs_print_latex
GiNaC, 150
abs_real_part
GiNaC, 150
accept
GiNaC::basic, 370
GiNaC::ex, 576
access_counter
GiNaC::remember_table_entry, 1100
acos
GiNaC, 239
acos_conjugate
GiNaC, 193
acos_deriv
GiNaC, 193
acos_eval
GiNaC, 192
acos_evalf
GiNaC, 192
acosh
GiNaC, 242
acosh_conjugate
GiNaC, 202
acosh_deriv
GiNaC, 202
acosh_eval
GiNaC, 201
acosh_evalf
GiNaC, 201
adaptivesimpson
GiNaC, 204
add
GiNaC::add, 326, 327
GiNaC::Gt_detail::pathintegral_term_list, 965
GiNaC::matrix, 825
GiNaC::mul, 871
GiNaC::numeric, 940
GiNaC::scalar_products, 1106
GiNaC::symmetry, 1196
add.cpp, 1261
add.h, 1262
add_bool
GiNaC::archive_node, 348
add_callback
GiNaC::_numeric_digits, 318
add_child
GiNaC::class_info< OPT >::tree_node, 1233
add_dyn
GiNaC::numeric, 942
add_entry
GiNaC::remember_table, 1096
GiNaC::remember_table_list, 1102
add_ex
GiNaC::archive_node, 349
add_indexed
GiNaC::basic, 376
GiNaC::matrix, 822
GiNaC::structure< T, ComparisonPolicy >, 1144
add_node
GiNaC::archive, 341
add_reference
GiNaC::refcounted, 1078
add_series
GiNaC::pseries, 1059
add_string
GiNaC::archive_node, 349
add_symbol
GiNaC, 220
add_unsigned

GiNaC::archive_node, 349
add_vectors
GiNaC::scalar_products, 1106
after_i
GiNaC::composition_generator::coolmulti, 486
algebraic
GiNaC::has_options, 727
GiNaC::subs_options, 1169
algebraic_match_mul_with_mul
GiNaC, 218
algebraic_subs_mul
GiNaC::mul, 869
all
GiNaC::factor_options, 625
all_index_values_are
GiNaC::indexed, 756
antisymmetric
GiNaC::symmetry, 1194
antisymmetric2
GiNaC, 276
antisymmetric3
GiNaC, 276
antisymmetric4
GiNaC, 276
antisymmetrize
GiNaC, 126, 277, 281
GiNaC::ex, 589
append
GiNaC::container< C >, 478
append_factors
GiNaC::ncmul, 922
apply_function_recursive
GiNaC::Gt, 721
applyIntegrationPath
GiNaC::Gt, 718
archive
GiNaC::archive, 338, 339
GiNaC::basic, 380
GiNaC::clifford, 414
GiNaC::color, 434
GiNaC::constant, 462
GiNaC::container< C >, 475
GiNaC::expairseq, 613
GiNaC::fderivative, 640
GiNaC::function, 659
GiNaC::Gt, 716
GiNaC::idx, 735
GiNaC::indexed, 754
GiNaC::integral, 770
GiNaC::matrix, 823
GiNaC::minkmetric, 842
GiNaC::numeric, 939
GiNaC::power, 1007
GiNaC::pseries, 1056
GiNaC::relational, 1089
GiNaC::spinidx, 1117
GiNaC::symbol, 1183
GiNaC::symmetry, 1195
GiNaC::tensepsilon, 1214
GiNaC::varidx, 1249
GiNaC::wildcard, 1257
archive.cpp, 1262
archive.h, 1263
GINAC_BIND_UNARCHIVER, 1265
GINAC_DECLARE_UNARCHIVER, 1264
archive_atom
GiNaC, 68
archive_ex
GiNaC::archive, 339
archive_node
GiNaC::archive_node, 348
GiNaC::ex, 593
archive_node_cit
GiNaC::archive_node, 347
archive_node_id
GiNaC, 68
archived_ex
GiNaC::archive::archived_ex, 357
are_ex_trivially_equal
GiNaC, 119
GiNaC::ex, 593
arg1
GiNaC::pointer_to_map_function_1arg< T1 >, 970
GiNaC::pointer_to_map_function_2args< T1, T2 >, 973
GiNaC::pointer_to_map_function_3args< T1, T2, T3 >, 975
GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >, 981
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >, 984
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >, 987
arg2
GiNaC::pointer_to_map_function_2args< T1, T2 >, 973
GiNaC::pointer_to_map_function_3args< T1, T2, T3 >, 975
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >, 985
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >, 988
arg3
GiNaC::pointer_to_map_function_3args< T1, T2, T3 >, 976
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >, 988
args
GiNaC::Gt, 723
argument_type
GiNaC::map_function, 813
asin
GiNaC, 239
asin_conjugate
GiNaC, 192

asin_deriv
GiNaC, 191
asin_eval
GiNaC, 191
asin_evalf
GiNaC, 191
asin_info
GiNaC, 192
asinh
GiNaC, 242
asinh_conjugate
GiNaC, 201
asinh_deriv
GiNaC, 201
asinh_eval
GiNaC, 200
asinh_evalf
GiNaC, 200
assertion.h, 1265
GINAC_ASSERT, 1266
atan
GiNaC, 240
atan2_deriv
GiNaC, 195
atan2_eval
GiNaC, 195
atan2_evalf
GiNaC, 195
atan2_info
GiNaC, 195
atan_conjugate
GiNaC, 194
atan_deriv
GiNaC, 194
atan_eval
GiNaC, 193
atan_evalf
GiNaC, 193
atan_info
GiNaC, 194
atan_series
GiNaC, 194
atanh
GiNaC, 242
atanh_conjugate
GiNaC, 203
atanh_deriv
GiNaC, 203
atanh_eval
GiNaC, 202
atanh_evalf
GiNaC, 202
atanh_series
GiNaC, 203
atend
GiNaC::composition_generator, 441
atomize
GiNaC::archive, 342
atoms
GiNaC::archive, 343
attribute_pure
compiler.h, 1278
auto_clear_polylog_cache
GiNaC::Gt, 723
automatic
GiNaC::determinant_algo, 489
GiNaC::solve_algo, 1109

B
GiNaC::basic_multi_iterator< T >, 399
b
GiNaC::Eisenstein_kernel, 545
GiNaC::integral, 772
bareiss
GiNaC::determinant_algo, 490
GiNaC::solve_algo, 1110
base_and_index
GiNaC, 100
basic
GiNaC::basic, 363
basic.cpp, 1267
basic.h, 1268
basic_multi_iterator
GiNaC::basic_multi_iterator< T >, 396
basic_partition_generator
GiNaC::basic_partition_generator, 402
basis
GiNaC::power, 1012
begin
GiNaC::archive_node::archive_node_cit_range, 356
GiNaC::container< C >, 479
GiNaC::ex, 568
GiNaC::Gt_detail::pathintegral_term_list, 964
bernoulli
GiNaC, 247
Bernoulli_polynomial
GiNaC, 206
beta_deriv
GiNaC, 173
beta_eval
GiNaC, 173
beta_evalf
GiNaC, 173
beta_series
GiNaC, 173
binomial
GiNaC, 246
binomial_conjugate
GiNaC, 160
binomial_eval
GiNaC, 160
binomial_evalf
GiNaC, 159
binomial_imag_part
GiNaC, 160
binomial_real_part

GiNaC, 160
binomial_sym
GiNaC, 160
bp
GiNaC::ex, 595

c
GiNaC::pointer_to_member_to_map_function< C >, 978
GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >, 981
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >, 984
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >, 987
C_norm
GiNaC::Eisenstein_h_kernel, 535
GiNaC::Eisenstein_kernel, 545
GiNaC::Kronecker_dtau_kernel, 795
GiNaC::Kronecker_dz_kernel, 804
GiNaC::modular_form_kernel, 851
cache
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1232
cache_step_size
GiNaC::integration_kernel, 782
calc_lanczos_A
GiNaC::lanczos_coeffs, 805
calchash
GiNaC::basic, 379
GiNaC::constant, 463
GiNaC::expairseq, 614
GiNaC::function, 657
GiNaC::Gt, 715
GiNaC::idx, 736
GiNaC::numeric, 940
GiNaC::relational, 1091
GiNaC::structure< T, ComparisonPolicy >, 1146
GiNaC::symbol, 1184
GiNaC::symmetry, 1195
GiNaC::wildcard, 1257
callback_registered
GiNaC, 92
callbacklist
GiNaC::_numeric_digits, 319
can_be_further_expanded
GiNaC::mul, 870
can_make_flat
GiNaC::expairseq, 617
GiNaC::mul, 868
canonical
GiNaC::relational, 1090
canonicalize
GiNaC, 276
GiNaC::expairseq, 619
GiNaC::symmetry, 1198
canonicalize_clifford
GiNaC, 105
Catalan
GiNaC, 294
CatalanEvalf
GiNaC, 252
charpoly
GiNaC, 216
GiNaC::matrix, 829
check
GiNaC::Gt::kernel, 786
check_type
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1231
children
GiNaC::class_info< OPT >::tree_node, 1233
GiNaC::symmetry, 1198
cinteger
GiNaC::info_flags, 760
cinteger_polynomial
GiNaC::info_flags, 761
class_info
GiNaC::class_info< OPT >, 404
class_info< OPT >::first
GiNaC, 293
class_info< OPT >::parents_identified
GiNaC, 293
class_info.h, 1269
clear
GiNaC::archive, 341
GiNaC::Gt_detail::pathintegral_term_list, 964
GiNaC::scalar_products, 1107
clear_all_entries
GiNaC::remember_table, 1096
clear_cache
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1231
clear_polylog_cache
GiNaC::Gt, 721
clearflag
GiNaC::basic, 384
clifford
GiNaC::clifford, 413, 414
clifford.cpp, 1270
clifford.h, 1272
clifford_bar
GiNaC, 112
clifford_inverse
GiNaC, 107
clifford_max_label
GiNaC, 106
clifford_moebius_map
GiNaC, 109, 110
clifford_norm
GiNaC, 107
clifford_prime
GiNaC, 106
clifford_star
GiNaC, 112
clifford_star_bar
GiNaC, 105

clifford_to_lst
GiNaC, 109
clifford_unit
GiNaC, 101
cmgen
GiNaC::composition_generator, 441
CMPINDICES
color.cpp, 1275
coeff
GiNaC, 122
GiNaC::add, 328
GiNaC::basic, 371
GiNaC::ex, 578
GiNaC::expair, 601
GiNaC::mul, 861
GiNaC::ncmul, 919
GiNaC::numeric, 935
GiNaC::power, 1003
GiNaC::pseries, 1053
GiNaC::structure< T, ComparisonPolicy >, 1140
GiNaC::symminfo, 1200
coefficient_a0
GiNaC::Eisenstein_h_kernel, 534
coefficient_an
GiNaC::Eisenstein_h_kernel, 534
coeffop
GiNaC::pseries, 1058
coeffs
GiNaC::lanczos_coeffs, 806
coerce
GiNaC, 236
coerce< int, cln::cl_I >
GiNaC, 236
coerce< unsigned int, cln::cl_I >
GiNaC, 236
col
GiNaC::matrix, 835
collect
GiNaC, 123
GiNaC::basic, 372
GiNaC::ex, 579
GiNaC::pseries, 1053
GiNaC::structure< T, ComparisonPolicy >, 1141
collect_common_factors
GiNaC, 233
collect_symbols
GiNaC, 220
color
GiNaC::color, 433, 434
color.cpp, 1274
CMPINDICES, 1275
TEST_PERMUTATION, 1275
color.h, 1276
color_d
GiNaC, 115
color_f
GiNaC, 115
color_h
GiNaC, 116
color_ONE
GiNaC, 114
color_T
GiNaC, 114
color_trace
GiNaC, 117
cols
GiNaC, 215
GiNaC::matrix, 824
combine_ex_with_coeff_to_pair
GiNaC::add, 334
GiNaC::expairseq, 616
GiNaC::mul, 867
combine_indices
GiNaC::make_flat_inserter, 810
combine_overall_coeff
GiNaC::expairseq, 617
GiNaC::mul, 868
combine_pair_with_coeff_to_pair
GiNaC::add, 334
GiNaC::expairseq, 616
GiNaC::mul, 867
combine_same_terms_sorted_seq
GiNaC::expairseq, 620
commutative
GiNaC::return_types, 1105
commutator_sign
GiNaC::clifford, 419
compare
GiNaC::basic, 382
GiNaC::ex, 587
GiNaC::expair, 600
GiNaC::numeric, 945
compare_pointers
GiNaC, 287
compare_same_type
GiNaC::basic, 379
compile_ex
GiNaC, 130, 131
compiler.h, 1277
attribute_pure, 1278
likely, 1278
unlikely, 1278
complex
GiNaC::domain, 516
composition
GiNaC::composition_generator, 442
composition_generator
GiNaC::composition_generator, 441
conjugate
GiNaC, 120
GiNaC::add, 331
GiNaC::basic, 378
GiNaC::constant, 461
GiNaC::container< C >, 475
GiNaC::diracgamma5, 500
GiNaC::diracgammaL, 505

GiNaC::diracgammaR, 510
GiNaC::ex, 573
GiNaC::expair, 601
GiNaC::expairseq, 612
GiNaC::function, 659
GiNaC::integral, 770
GiNaC::matrix, 823
GiNaC::mul, 865
GiNaC::ncmul, 920
GiNaC::numeric, 939
GiNaC::power, 1007
GiNaC::pseries, 1055
GiNaC::realsymbol, 1075
GiNaC::spinidx, 1117
GiNaC::symbol, 1182
conjugate_conjugate
GiNaC, 144
conjugate_eval
GiNaC, 144
conjugate_evalf
GiNaC, 143
conjugate_expl_derivative
GiNaC, 144
conjugate_f
GiNaC::function_options, 701
conjugate_func
GiNaC::function_options, 676–678, 695
conjugate_funcp
GiNaC, 70
conjugate_funcp_1
GiNaC, 71
conjugate_funcp_10
GiNaC, 83
conjugate_funcp_11
GiNaC, 84
conjugate_funcp_12
GiNaC, 85
conjugate_funcp_13
GiNaC, 87
conjugate_funcp_14
GiNaC, 88
conjugate_funcp_2
GiNaC, 72
conjugate_funcp_3
GiNaC, 74
conjugate_funcp_4
GiNaC, 75
conjugate_funcp_5
GiNaC, 76
conjugate_funcp_6
GiNaC, 77
conjugate_funcp_7
GiNaC, 79
conjugate_funcp_8
GiNaC, 80
conjugate_funcp_9
GiNaC, 81
conjugate_funcp_exvector
GiNaC, 90
conjugate_imag_part
GiNaC, 145
conjugate_info
GiNaC, 145
conjugate_print_latex
GiNaC, 144
conjugate_real_part
GiNaC, 144
conjugate_use_exvector_args
GiNaC::function_options, 704
conjugateepvector
GiNaC, 133
const_iterator
GiNaC::const_iterator, 445
GiNaC::container< C >, 472
const_postorder_iterator
GiNaC::const_iterator, 448
GiNaC::const_postorder_iterator, 450
const_preorder_iterator
GiNaC::const_iterator, 447
GiNaC::const_preorder_iterator, 453
const_reverse_iterator
GiNaC::container< C >, 472
constant
GiNaC::constant, 460
constant.cpp, 1278
constant.h, 1279
construct_from_2_ex
GiNaC::expairseq, 618
construct_from_2_expairseq
GiNaC::expairseq, 618
construct_from_basic
GiNaC::ex, 590
construct_from_double
GiNaC::ex, 592
construct_from_epvector
GiNaC::expairseq, 619
construct_from_expairseq_ex
GiNaC::expairseq, 618
construct_from_exvector
GiNaC::expairseq, 618
construct_from_int
GiNaC::ex, 591
construct_from_long
GiNaC::ex, 591
construct_from_longlong
GiNaC::ex, 592
construct_from_string_and_lst
GiNaC::ex, 592
construct_from_uint
GiNaC::ex, 591
construct_from_ulong
GiNaC::ex, 591
construct_from_ulonglong
GiNaC::ex, 592
container
GiNaC::container< C >, 472

container.h, 1280
container_storage
GiNaC::container_storage< C >, 482, 483
content
GiNaC::ex, 583
contract_with
GiNaC::basic, 377
GiNaC::cliffordunit, 425
GiNaC::diracgamma, 495
GiNaC::matrix, 822
GiNaC::spinmetric, 1125
GiNaC::structure< T, ComparisonPolicy >, 1145
GiNaC::su3d, 1152
GiNaC::su3f, 1158
GiNaC::su3t, 1168
GiNaC::tensdelta, 1207
GiNaC::tensepsilon, 1213
GiNaC::tensmetric, 1221
convert_H_to_Li
GiNaC, 166
convert_to_poly
GiNaC::pseries, 1058
coolmulti
GiNaC::composition_generator::coolmulti, 485
cos
GiNaC, 238
cos_conjugate
GiNaC, 189
cos_deriv
GiNaC, 188
cos_eval
GiNaC, 188
cos_evalf
GiNaC, 188
cos_imag_part
GiNaC, 189
cos_real_part
GiNaC, 189
cosh
GiNaC, 241
cosh_conjugate
GiNaC, 198
cosh_deriv
GiNaC, 198
cosh_eval
GiNaC, 197
cosh_evalf
GiNaC, 197
cosh_imag_part
GiNaC, 198
cosh_real_part
GiNaC, 198
count
GiNaC::archive_node::property_info, 1045
GiNaC::library_init, 809
count_dummy_indices
GiNaC, 138
count_factors
GiNaC::ncmul, 922
count_free_indices
GiNaC, 138
covariant
GiNaC::varidx, 1251
crational
GiNaC::info_flags, 760
crational_polynomial
GiNaC::info_flags, 761
crc32
GiNaC, 119
crc32.h, 1280
crctab
GiNaC, 294
csgn
GiNaC, 254
GiNaC::numeric, 945
csgn_conjugate
GiNaC, 153
csgn_eval
GiNaC, 153
csgn_evalf
GiNaC, 153
csgn_imag_part
GiNaC, 154
csgn_power
GiNaC, 154
csgn_real_part
GiNaC, 154
csgn_series
GiNaC, 153
csrc
GiNaC, 267
csrc_cl_N
GiNaC, 268
csrc_double
GiNaC, 268
csrc_float
GiNaC, 268
current_serial
GiNaC::function, 664
current_updated
GiNaC::composition_generator, 442
GiNaC::partition_generator, 957
GiNaC::partition_with_zero_parts_generator, 959
current_vector
GiNaC::lanczos_coeffs, 806
cutIntegrationPath_threshold_imag
GiNaC::Gt, 723
cutIntegrationPath_threshold_real
GiNaC::Gt, 723
cyclic
GiNaC::symmetry, 1194
cyclic_permutation
GiNaC, 289

dbgprint
GiNaC::basic, 366
GiNaC::ex, 570

dbgprinttree
GiNaC::basic, 366
GiNaC::ex, 571
DCOUT
factor.cpp, 1290
DCOUT2
factor.cpp, 1290
DCOUTVAR
factor.cpp, 1290
DEBUG_GT
Gt.cpp, 1308
debugprint
GiNaC::scalar_products, 1107
GiNaC::spmapkey, 1128
DECLARE_FUNCTION_10P
function.h, 1304
DECLARE_FUNCTION_11P
function.h, 1304
DECLARE_FUNCTION_12P
function.h, 1304
DECLARE_FUNCTION_13P
function.h, 1305
DECLARE_FUNCTION_14P
function.h, 1305
DECLARE_FUNCTION_1P
function.h, 1302
DECLARE_FUNCTION_2P
function.h, 1302
DECLARE_FUNCTION_3P
function.h, 1302
DECLARE_FUNCTION_4P
function.h, 1302
DECLARE_FUNCTION_5P
function.h, 1303
DECLARE_FUNCTION_6P
function.h, 1303
DECLARE_FUNCTION_7P
function.h, 1303
DECLARE_FUNCTION_8P
function.h, 1303
DECLARE_FUNCTION_9P
function.h, 1304
decomp_rational
GiNaC, 223
decomposeIntegrationPath
GiNaC::Gt, 717
deconcatenate_path
GiNaC::Gt_detail, 312
deconcatenate_path< ex >
GiNaC::Gt_detail, 312
deconcatenate_path< Gt::kernel >
GiNaC::Gt_detail, 312
DEFAULT_COMPARE
utils.h, 1375
DEFAULT_CTOR
utils.h, 1375
default_overall_coeff
GiNaC::expairseq, 617
GiNaC::mul, 868
DEFAULT_PRINT
utils.h, 1375
DEFAULT_PRINT_LATEX
utils.h, 1376
deform
GiNaC::Gt::kernel, 787
deg
GiNaC::pole_error, 990
deg_a
GiNaC::sym_desc, 1174
deg_b
GiNaC::sym_desc, 1174
degree
GiNaC, 121
GiNaC::add, 328
GiNaC::basic, 371
GiNaC::ex, 577
GiNaC::integral, 768
GiNaC::mul, 861
GiNaC::ncmul, 918
GiNaC::numeric, 935
GiNaC::pole_error, 990
GiNaC::power, 1003
GiNaC::pseries, 1053
GiNaC::structure< T, ComparisonPolicy >, 1140
delete_cyclic
GiNaC::remember_strategies, 1094
delete_lfu
GiNaC::remember_strategies, 1094
delete_lru
GiNaC::remember_strategies, 1094
delete_never
GiNaC::remember_strategies, 1094
delta_indent
GiNaC::print_tree, 1042
delta_tensor
GiNaC, 283
denom
GiNaC, 122, 258
GiNaC::ex, 581
GiNaC::Gt_detail::pathintegral_term, 963
GiNaC::numeric, 952
derivative
GiNaC::add, 332
GiNaC::basic, 372
GiNaC::constant, 462
GiNaC::fderivative, 641
GiNaC::function, 660
GiNaC::idx, 736
GiNaC::indexed, 754
GiNaC::integral, 771
GiNaC::mul, 865
GiNaC::ncmul, 921
GiNaC::numeric, 940
GiNaC::power, 1008
GiNaC::pseries, 1057
GiNaC::structure< T, ComparisonPolicy >, 1141

GiNaC::symbol, 1184
derivative_f
GiNaC::function_options, 702
derivative_func
GiNaC::function_options, 684–686, 695
derivative_funcp
GiNaC, 70
derivative_funcp_1
GiNaC, 72
derivative_funcp_10
GiNaC, 83
derivative_funcp_11
GiNaC, 84
derivative_funcp_12
GiNaC, 86
derivative_funcp_13
GiNaC, 87
derivative_funcp_14
GiNaC, 89
derivative_funcp_2
GiNaC, 73
derivative_funcp_3
GiNaC, 74
derivative_funcp_4
GiNaC, 75
derivative_funcp_5
GiNaC, 76
derivative_funcp_6
GiNaC, 78
derivative_funcp_7
GiNaC, 79
derivative_funcp_8
GiNaC, 80
derivative_funcp_9
GiNaC, 82
derivative_funcp_exvector
GiNaC, 90
derivative_map_function
GiNaC::derivative_map_function, 488
derivative_use_exvector_args
GiNaC::function_options, 704
derivatives
GiNaC::fderivative, 642
descend
GiNaC::const_postorder_iterator, 451
determinant
GiNaC, 215
GiNaC::matrix, 828
determinant_minor
GiNaC::matrix, 831
dflt
GiNaC, 267
diag_matrix
GiNaC, 213
diff
GiNaC, 124
GiNaC::basic, 382
GiNaC::ex, 579
difference_type
GiNaC::const_iterator, 444
GiNaC::const_postorder_iterator, 449
GiNaC::const_preorder_iterator, 452
Digits
GiNaC, 296
digits
GiNaC::_numeric_digits, 319
digits_changed_callback
GiNaC, 92
dim
GiNaC::idx, 740
GiNaC::spmapkey, 1129
dirac_gamma
GiNaC, 101
dirac_gamma5
GiNaC, 101
dirac_gammaL
GiNaC, 102
dirac_gammaR
GiNaC, 102
dirac_ONE
GiNaC, 100
dirac_slash
GiNaC, 103
dirac_trace
GiNaC, 104, 105
dirichlet_character
GiNaC, 206
div
GiNaC::numeric, 941
div_dyn
GiNaC::numeric, 943
divfree
GiNaC::determinant_algo, 489
GiNaC::solve_algo, 1109
divide
GiNaC, 224
divide_in_z
GiNaC, 225
division_free_elimination
GiNaC::matrix, 832
do_not_evalf_params
GiNaC::function_options, 699
do_print
GiNaC::add, 335
GiNaC::basic, 385
GiNaC::basic_log_kernel, 393
GiNaC::cliffordunit, 425
GiNaC::constant, 464
GiNaC::container< C >, 480
GiNaC::diracgamma, 495
GiNaC::diracgamma5, 500
GiNaC::diracgammaL, 505
GiNaC::diracgammaR, 510
GiNaC::diracone, 515
GiNaC::Ebar_kernel, 524
GiNaC::Eisenstein_h_kernel, 534

GiNaC::Eisenstein_kernel, 544
GiNaC::ELi_kernel, 554
GiNaC::expairseq, 617
GiNaC::fail, 630
GiNaC::fderivative, 642
GiNaC::Gt, 722
GiNaC::idx, 739
GiNaC::indexed, 757
GiNaC::integral, 771
GiNaC::integration_kernel, 781
GiNaC::Kronecker_dtau_kernel, 794
GiNaC::Kronecker_dz_kernel, 803
GiNaC::matrix, 834
GiNaC::minkmetric, 842
GiNaC::modular_form_kernel, 851
GiNaC::mul, 869
GiNaC::multiple_polylog_kernel, 910
GiNaC::ncmul, 922
GiNaC::numeric, 952
GiNaC::pseries, 1061
GiNaC::relational, 1091
GiNaC::spinidx, 1119
GiNaC::spinmetric, 1126
GiNaC::su3d, 1153
GiNaC::su3f, 1158
GiNaC::su3one, 1163
GiNaC::su3t, 1168
GiNaC::symbol, 1186
GiNaC::symmetry, 1197
GiNaC::tensdelta, 1207
GiNaC::tensepsilon, 1214
GiNaC::tensmetric, 1221
GiNaC::user_defined_kernel, 1242
GiNaC::varidx, 1251
GiNaC::wildcard, 1258
do_print_csrc
GiNaC::add, 335
GiNaC::fderivative, 643
GiNaC::idx, 739
GiNaC::mul, 870
GiNaC::ncmul, 922
GiNaC::numeric, 953
GiNaC::power, 1010
do_print_csrc_cl_N
GiNaC::numeric, 953
GiNaC::power, 1010
do_print_dflt
GiNaC::clifford, 418
GiNaC::power, 1009
do_print_latex
GiNaC::add, 335
GiNaC::clifford, 419
GiNaC::cliffordunit, 425
GiNaC::constant, 464
GiNaC::diracgamma, 495
GiNaC::diracgamma5, 500
GiNaC::diracgammaL, 505
GiNaC::diracgammaR, 510
GiNaC::diracone, 515
GiNaC::fderivative, 643
GiNaC::Gt, 722
GiNaC::idx, 739
GiNaC::indexed, 757
GiNaC::integral, 772
GiNaC::matrix, 835
GiNaC::minkmetric, 842
GiNaC::mul, 870
GiNaC::numeric, 953
GiNaC::power, 1009
GiNaC::pseries, 1061
GiNaC::spinidx, 1119
GiNaC::spinmetric, 1126
GiNaC::su3d, 1153
GiNaC::su3f, 1158
GiNaC::su3one, 1163
GiNaC::su3t, 1168
GiNaC::symbol, 1186
GiNaC::tensdelta, 1208
GiNaC::tensepsilon, 1215
do_print_python
GiNaC::container< C >, 480
GiNaC::power, 1010
GiNaC::pseries, 1061
do_print_python_repr
GiNaC::add, 336
GiNaC::basic, 385
GiNaC::constant, 464
GiNaC::container< C >, 480
GiNaC::matrix, 835
GiNaC::mul, 870
GiNaC::numeric, 953
GiNaC::power, 1010
GiNaC::pseries, 1062
GiNaC::relational, 1091
GiNaC::symbol, 1186
GiNaC::wildcard, 1258
do_print_tree
GiNaC::basic, 385
GiNaC::clifford, 419
GiNaC::constant, 464
GiNaC::container< C >, 480
GiNaC::expairseq, 618
GiNaC::fderivative, 643
GiNaC::idx, 739
GiNaC::indexed, 758
GiNaC::numeric, 953
GiNaC::pseries, 1061
GiNaC::spinidx, 1119
GiNaC::symbol, 1186
GiNaC::symmetry, 1197
GiNaC::varidx, 1251
GiNaC::wildcard, 1258
do_renaming
GiNaC::make_flat_inserter, 810
domain
GiNaC::constant, 465

dotted
GiNaC::spinidx, 1120
doublefactorial
GiNaC, 246
dummy
GiNaC::function_options, 672
dump_hierarchy
GiNaC::class_info< OPT >, 404
dump_tree
GiNaC::class_info< OPT >, 404
duplicate
GiNaC::basic, 364
GiNaC::possymbol, 996
GiNaC::print_functor_impl, 1029
GiNaC::print_memfun_handler< T, C >, 1033
GiNaC::print_ptrfun_handler< T, C >, 1037
GiNaC::realsymbol, 1075
dynallocate
GiNaC, 97, 98
dynallocated
GiNaC::status_flags, 1130

e
GiNaC::archive_node, 355
GiNaC::const_iterator, 448
GiNaC::internal::_iter_rep, 316
Ebar_kernel
GiNaC::Ebar_kernel, 523
echelon_form
GiNaC::matrix, 832
ef
GiNaC::constant, 465
Eisenstein_h_kernel
GiNaC::Eisenstein_h_kernel, 532
Eisenstein_kernel
GiNaC::Eisenstein_kernel, 542
element
GiNaC::composition_generator::coolmulti::element, 546
ELi_kernel
GiNaC::ELi_kernel, 553
EllipticE_deriv
GiNaC, 168
EllipticE_eval
GiNaC, 168
EllipticE_evalf
GiNaC, 167
EllipticE_print_latex
GiNaC, 168
EllipticE_series
GiNaC, 168
EllipticK_deriv
GiNaC, 167
EllipticK_eval
GiNaC, 166
EllipticK_evalf
GiNaC, 166
EllipticK_print_latex
GiNaC, 167
EllipticK_series
GiNaC, 167
empty
GiNaC::Gt_detail::pathintegral_term_list, 964
enable_tauToFundamental
GiNaC::Gt, 723
end
GiNaC::archive_node::archive_node_cit_range, 356
GiNaC::container< C >, 479
GiNaC::ex, 568
GiNaC::Gt_detail::pathintegral_term_list, 964
ensure_if_modifiable
GiNaC::basic, 384
epp
GiNaC, 69
epsilon_tensor
GiNaC, 284, 285
epvector
GiNaC, 69
equal
GiNaC::relational, 1087
error
GiNaC::error_and_integral, 557
error_and_integral
GiNaC::error_and_integral, 557
eta_conjugate
GiNaC, 155
eta_eval
GiNaC, 155
eta_evalf
GiNaC, 154
eta_imag_part
GiNaC, 155
eta_real_part
GiNaC, 155
eta_series
GiNaC, 155
Euler
GiNaC, 294
EulerEvalf
GiNaC, 252
eval
GiNaC, 123
GiNaC::add, 329
GiNaC::basic, 364
GiNaC::ex, 569
GiNaC::expairseq, 611
GiNaC::fderivative, 639
GiNaC::function, 657
GiNaC::Gt, 714
GiNaC::indexed, 753
GiNaC::integral, 767
GiNaC::mul, 862
GiNaC::ncmul, 919
GiNaC::numeric, 936
GiNaC::power, 1003
GiNaC::pseries, 1054

GiNaC::structure< T, ComparisonPolicy >, 1135
GiNaC::symbol, 1181
eval_f
GiNaC::function_options, 701
eval_func
GiNaC::function_options, 672–674, 694
eval_funcp
GiNaC, 70
eval_funcp_1
GiNaC, 71
eval_funcp_10
GiNaC, 82
eval_funcp_11
GiNaC, 84
eval_funcp_12
GiNaC, 85
eval_funcp_13
GiNaC, 87
eval_funcp_14
GiNaC, 88
eval_funcp_2
GiNaC, 72
eval_funcp_3
GiNaC, 73
eval_funcp_4
GiNaC, 75
eval_funcp_5
GiNaC, 76
eval_funcp_6
GiNaC, 77
eval_funcp_7
GiNaC, 78
eval_funcp_8
GiNaC, 80
eval_funcp_9
GiNaC, 81
eval_funcp_exvector
GiNaC, 90
eval_indexed
GiNaC::basic, 365
GiNaC::matrix, 822
GiNaC::minkmetric, 841
GiNaC::spinmetric, 1125
GiNaC::structure< T, ComparisonPolicy >, 1136
GiNaC::su3d, 1152
GiNaC::su3f, 1158
GiNaC::tensdelta, 1207
GiNaC::tensepsilon, 1213
GiNaC::tensmetric, 1221
eval_integ
GiNaC, 124
GiNaC::basic, 365
GiNaC::ex, 570
GiNaC::integral, 770
GiNaC::pseries, 1056
eval_ncmul
GiNaC::add, 332
GiNaC::basic, 365
GiNaC::clifford, 415
GiNaC::color, 434
GiNaC::ex, 570
GiNaC::function, 657
GiNaC::integral, 768
GiNaC::mul, 865
GiNaC::power, 1008
GiNaC::relational, 1090
GiNaC::structure< T, ComparisonPolicy >, 1136
eval_use_exvector_args
GiNaC::function_options, 704
evalchildren
GiNaC::expairseq, 620
evalf
GiNaC, 124, 215
GiNaC::basic, 364
GiNaC::constant, 461
GiNaC::ex, 569
GiNaC::function, 657
GiNaC::Gt, 715
GiNaC::idx, 734
GiNaC::integral, 767
GiNaC::mul, 862
GiNaC::numeric, 936
GiNaC::power, 1004
GiNaC::pseries, 1054
GiNaC::symbol, 1181
evalf_f
GiNaC::function_options, 701
evalf_func
GiNaC::function_options, 674–676, 694
evalf_funcp
GiNaC, 70
evalf_funcp_1
GiNaC, 71
evalf_funcp_10
GiNaC, 83
evalf_funcp_11
GiNaC, 84
evalf_funcp_12
GiNaC, 85
evalf_funcp_13
GiNaC, 87
evalf_funcp_14
GiNaC, 88
evalf_funcp_2
GiNaC, 72
evalf_funcp_3
GiNaC, 73
evalf_funcp_4
GiNaC, 75
evalf_funcp_5
GiNaC, 76
evalf_funcp_6
GiNaC, 77
evalf_funcp_7
GiNaC, 79
evalf_funcp_8

GiNaC, 80
evalf_funcp_9
GiNaC, 81
evalf_funcp_exvector
GiNaC, 90
evalf_params_first
GiNaC::function_options, 703
evalf_use_exvector_args
GiNaC::function_options, 704
evalffunctype
GiNaC, 69
evalm
GiNaC, 124
GiNaC::add, 329
GiNaC::basic, 364
GiNaC::ex, 569
GiNaC::matrix, 821
GiNaC::mul, 863
GiNaC::ncmul, 919
GiNaC::power, 1004
GiNaC::pseries, 1056
GiNaC::structure< T, ComparisonPolicy >, 1136
evaluate
GiNaC::Gt, 718
GiNaC::Gt_detail::pathintegral_term, 962
GiNaC::scalar_products, 1107
evaluated
GiNaC::status_flags, 1130
even
GiNaC::info_flags, 760
ex
GiNaC::basic, 386
GiNaC::const_iterator, 447
GiNaC::ex, 566–568
ex.cpp, 1281
ex.h, 1281
ex_cutIntegrationPath
GiNaC::Gt, 719, 720
ex_evaluate
GiNaC::Gt, 719, 721
ex_prepare
GiNaC::Gt, 719, 721
ex_qExpand
GiNaC::Gt, 719, 721
ex_regularise
GiNaC::Gt, 718, 720
ex_tauToFundamental
GiNaC::Gt, 719, 720
ex_to
GiNaC, 128
GiNaC::ex, 593
ex_zisToFundamental
GiNaC::Gt, 718, 720
exadd
GiNaC, 258
excompiler.cpp, 1284
excompiler.h, 1284
exhashmap
GiNaC, 91
exmap
GiNaC, 69
exminus
GiNaC, 258
exmul
GiNaC, 258
exp
GiNaC, 237
exp_conjugate
GiNaC, 184
exp_deriv
GiNaC, 183
exp_eval
GiNaC, 183
exp_evalf
GiNaC, 182
exp_expand
GiNaC, 183
exp_imag_part
GiNaC, 183
exp_info
GiNaC, 184
exp_power
GiNaC, 184
exp_real_part
GiNaC, 183
expair
GiNaC::expair, 599
expair.cpp, 1285
expair.h, 1286
expair_needs_further_processing
GiNaC::expairseq, 616
GiNaC::mul, 867
expairseq
GiNaC::expairseq, 609, 610
expairseq.cpp, 1287
expairseq.h, 1287
expand
GiNaC, 120, 215
GiNaC::add, 334
GiNaC::basic, 372
GiNaC::ex, 578
GiNaC::expairseq, 614
GiNaC::function, 656
GiNaC::indexed, 755
GiNaC::integral, 769
GiNaC::mul, 868
GiNaC::ncmul, 918
GiNaC::power, 1009
GiNaC::pseries, 1055
GiNaC::structure< T, ComparisonPolicy >, 1141
expand_add
GiNaC::power, 1010
expand_add_2
GiNaC::power, 1011
expand_dummy_sum
GiNaC, 143

expand_f
GiNaC::function_options, 702
expand_func
GiNaC::function_options, 682–684, 695
expand_funcp
GiNaC, 70
expand_funcp_1
GiNaC, 71
expand_funcp_10
GiNaC, 83
expand_funcp_11
GiNaC, 84
expand_funcp_12
GiNaC, 86
expand_funcp_13
GiNaC, 87
expand_funcp_14
GiNaC, 89
expand_funcp_2
GiNaC, 73
expand_funcp_3
GiNaC, 74
expand_funcp_4
GiNaC, 75
expand_funcp_5
GiNaC, 76
expand_funcp_6
GiNaC, 78
expand_funcp_7
GiNaC, 79
expand_funcp_8
GiNaC, 80
expand_funcp_9
GiNaC, 82
expand_funcp_exvector
GiNaC, 90
expand_function_args
GiNaC::expand_options, 625
expand_indexed
GiNaC::expand_options, 625
expand_map_function
GiNaC::expand_map_function, 623
expand_mul
GiNaC::power, 1011
expand_rename_idx
GiNaC::expand_options, 625
expand_transcendental
GiNaC::expand_options, 625
expand_use_exvector_args
GiNaC::function_options, 704
expandchildren
GiNaC::expairseq, 620
GiNaC::mul, 870
GiNaC::ncmul, 922
expanded
GiNaC::info_flags, 761
GiNaC::status_flags, 1130
expl_derivative
GiNaC::function, 662
expl_derivative_f
GiNaC::function_options, 702
expl_derivative_func
GiNaC::function_options, 686–688, 695
expl_derivative_funcp
GiNaC, 70
expl_derivative_funcp_1
GiNaC, 72
expl_derivative_funcp_10
GiNaC, 83
expl_derivative_funcp_11
GiNaC, 85
expl_derivative_funcp_12
GiNaC, 86
expl_derivative_funcp_13
GiNaC, 88
expl_derivative_funcp_14
GiNaC, 89
expl_derivative_funcp_2
GiNaC, 73
expl_derivative_funcp_3
GiNaC, 74
expl_derivative_funcp_4
GiNaC, 75
expl_derivative_funcp_5
GiNaC, 77
expl_derivative_funcp_6
GiNaC, 78
expl_derivative_funcp_7
GiNaC, 79
expl_derivative_funcp_8
GiNaC, 81
expl_derivative_funcp_9
GiNaC, 82
expl_derivative_funcp_exvector
GiNaC, 90
expl_derivative_use_exvector_args
GiNaC::function_options, 704
exponent
GiNaC::power, 1012
exponop
GiNaC::pseries, 1059
exprs
GiNaC::archive, 343
exprseq
GiNaC, 70
GiNaC::info_flags, 761
exprseq.cpp, 1288
exprseq.h, 1289
exprtable
GiNaC::archive, 344
exset
GiNaC, 69
exvector
GiNaC, 69
exvectorvector
GiNaC, 91

F
GiNaC::print_memfun_handler< T, C >, 1033
GiNaC::print_ptrfun_handler< T, C >, 1036
f
GiNaC::integral, 773
GiNaC::print_memfun_handler< T, C >, 1034
GiNaC::print_ptrfun_handler< T, C >, 1037
GiNaC::user_defined_kernel, 1242
factor
GiNaC, 133
factor.cpp, 1289
DCOUT, 1290
DCOUT2, 1290
DCOUTVAR, 1290
USE_SAME_DEGREE_FACTOR, 1290
factor.h, 1291
factorial
GiNaC, 245
factorial_conjugate
GiNaC, 159
factorial_eval
GiNaC, 158
factorial_evalf
GiNaC, 158
factorial_imag_part
GiNaC, 159
factorial_print_dflt_latex
GiNaC, 158
factorial_real_part
GiNaC, 159
fail.cpp, 1291
fail.h, 1292
FAST_COMPARE
normal.cpp, 1341
fderivative
GiNaC::fderivative, 638, 639
GiNaC::function_options, 700
fderivative.cpp, 1292
fderivative.h, 1293
fibonacci
GiNaC, 247
find
GiNaC, 121
GiNaC::class_info< OPT >, 404
GiNaC::ex, 574
GiNaC::unarchive_table_t, 1234
find_bool
GiNaC::archive_node, 349
find_common_factor
GiNaC, 233
find_dummy_indices
GiNaC, 138
find_ex
GiNaC::archive_node, 351
find_ex_by_loc
GiNaC::archive_node, 351
find_ex_node
GiNaC::archive_node, 352
find_factory_fcn
GiNaC, 95
find_first
GiNaC::archive_node, 350
find_free_and_dummy
GiNaC, 136, 138
find_function
GiNaC::function, 663
find_last
GiNaC::archive_node, 351
find_property_range
GiNaC::archive_node, 351
find_real_imag
GiNaC::mul, 869
find_string
GiNaC::archive_node, 350
find_unsigned
GiNaC::archive_node, 350
find_variant_indices
GiNaC, 140
findMoebiusTransform
GiNaC::Gt, 717
finished
GiNaC::composition_generator::coolmulti, 486
first
GiNaC::class_info< OPT >, 405
flag_overflow
GiNaC::basic_multi_iterator< T >, 400
flags
GiNaC::basic, 387
flags.h, 1293
force_include_tgamma
GiNaC, 295
force_include_zeta1
GiNaC, 295
forget
GiNaC::archive, 341
GiNaC::archive_node, 353
format_index_value
GiNaC, 289
frac_cancel
GiNaC, 233
fraction_free_elimination
GiNaC::matrix, 833
fsolve
GiNaC, 163
func_arg_info
GiNaC, 145
func_default
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1232
func_obj
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1232
FUNCP_1P
GiNaC, 69
FUNCP_2P
GiNaC, 69

FUNCP_CUBA
GiNaC, 69
function
GiNaC::function, 651–655
GiNaC::function_options, 700
function.cpp, 1294
function.h, 1295
DECLARE_FUNCTION_10P, 1304
DECLARE_FUNCTION_11P, 1304
DECLARE_FUNCTION_12P, 1304
DECLARE_FUNCTION_13P, 1305
DECLARE_FUNCTION_14P, 1305
DECLARE_FUNCTION_1P, 1302
DECLARE_FUNCTION_2P, 1302
DECLARE_FUNCTION_3P, 1302
DECLARE_FUNCTION_4P, 1302
DECLARE_FUNCTION_5P, 1303
DECLARE_FUNCTION_6P, 1303
DECLARE_FUNCTION_7P, 1303
DECLARE_FUNCTION_8P, 1303
DECLARE_FUNCTION_9P, 1304
is_ex_the_function, 1305
REGISTER_FUNCTION, 1305
function_options
GiNaC::function_options, 671
functions_with_same_name
GiNaC::function_options, 705

G
GiNaC, 164
G2_eval
GiNaC, 176
G2_evalf
GiNaC, 175
G3_eval
GiNaC, 176
G3_evalf
GiNaC, 176
G_path
GiNaC::Gt_detail::pathintegral_term, 962
gauss
GiNaC::determinant_algo, 489
GiNaC::solve_algo, 1109
gauss_elimination
GiNaC::matrix, 832
gcd
GiNaC, 229, 251
gcd_pf_mul
GiNaC, 228
gcd_pf_pow
GiNaC, 228
gcd_pf_pow_pow
GiNaC, 229
generalised_Bernoulli_number
GiNaC, 206
get
GiNaC::composition_generator, 441
GiNaC::partition_generator, 956
GiNaC::partition_with_zero_parts_generator, 959
get_all_dummy_indices
GiNaC, 142
get_all_dummy_indices_safely
GiNaC, 141
get_cache
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1231
get_cache_size
GiNaC::integration_kernel, 780
get_class_name
GiNaC::structure< T, ComparisonPolicy >, 1135
get_clifford_comp
GiNaC, 108
get_close_delim
GiNaC::container< C >, 473
get_commutator_sign
GiNaC::clifford, 417
get_default_flags
GiNaC::container< C >, 473
get_default_TeX_name
GiNaC, 273
get_dim
GiNaC::idx, 738
get_dim_uint
GiNaC, 100
get_domain
GiNaC::possymbol, 996
GiNaC::realsymbol, 1075
GiNaC::symbol, 1185
get_dummy_indices
GiNaC::indexed, 756
get_ex
GiNaC::archive_node, 353
get_factors
GiNaC::ncmul, 923
get_first_symbol
GiNaC, 219
get_free_indices
GiNaC::add, 332
GiNaC::basic, 375
GiNaC::ex, 586
GiNaC::indexed, 753
GiNaC::integral, 769
GiNaC::mul, 865
GiNaC::ncmul, 920
GiNaC::structure< T, ComparisonPolicy >, 1144
get_id
GiNaC::print_context_options, 1016
GiNaC::registered_class_options, 1080
get_indices
GiNaC::indexed, 756
get_label
GiNaC::wildcard, 1257
get_last_access
GiNaC::remember_table_entry, 1099
get_metric
GiNaC::clifford, 417
get_name

GiNaC::constant, 463
GiNaC::function, 663
GiNaC::function_options, 699
GiNaC::print_context_options, 1016
GiNaC::registered_class_options, 1080
GiNaC::symbol, 1185
get_node
GiNaC::archive, 341
get_nparams
GiNaC::function_options, 699
get_numerical_value
GiNaC::Ebar_kernel, 524
GiNaC::Eisenstein_h_kernel, 533
GiNaC::Eisenstein_kernel, 543
GiNaC::ELi_kernel, 554
GiNaC::integration_kernel, 779
GiNaC::Kronecker_dtau_kernel, 794
GiNaC::Kronecker_dz_kernel, 803
GiNaC::modular_form_kernel, 850
get_numerical_value_impl
GiNaC::integration_kernel, 781
get_open_delim
GiNaC::container< C >, 473
get_order
GiNaC::lanczos_coeffs, 805
get_parent
GiNaC::class_info< OPT >, 404
get_parent_name
GiNaC::print_context_options, 1016
GiNaC::registered_class_options, 1080
get_point
GiNaC::pseries, 1057
get_pointer
GiNaC::ptr< T >, 1068
get_print_context
GiNaC, 265
get_print_dispatch_table
GiNaC::registered_class_options, 1080
get_print_options
GiNaC, 266
get_properties
GiNaC::archive_node, 352
get_refcount
GiNaC::refcounted, 1078
get_registered_functions
GiNaC::function, 663
get_representation_label
GiNaC, 103, 116
GiNaC::clifford, 417
GiNaC::color, 436
get_result
GiNaC::remember_table_entry, 1099
get_serial
GiNaC::function, 663
get_series_coeff
GiNaC::integration_kernel, 780
get_sign
GiNaC::multi_iterator_permutation< T >, 894
get_struct
GiNaC::structure< T, ComparisonPolicy >, 1147
get_successful_hits
GiNaC::remember_table_entry, 1099
get_symbol_stats
GiNaC, 220
get_symmetry
GiNaC::indexed, 757
get_TeX_name
GiNaC::constant, 463
GiNaC::symbol, 1185
get_top_node
GiNaC::archive, 341
get_type
GiNaC::symmetry, 1195
get_value
GiNaC::idx, 737
get_var
GiNaC::pseries, 1057
get_vector
GiNaC::basic_multi_iterator< T >, 397
getArgs
GiNaC::Gt, 714
gethash
GiNaC::basic, 383
GiNaC::ex, 590
getTau
GiNaC::Gt, 714
getZ
GiNaC::Gt, 714
GiNaC, 29
_ex0, 303
_ex1, 305
_ex10, 308
_ex11, 308
_ex12, 309
_ex120, 311
_ex15, 309
_ex18, 309
_ex1_2, 305
_ex1_3, 304
_ex1_4, 304
_ex2, 306
_ex20, 310
_ex24, 310
_ex25, 310
_ex3, 306
_ex30, 310
_ex4, 306
_ex48, 311
_ex5, 307
_ex6, 307
_ex60, 311
_ex7, 307
_ex8, 308
_ex9, 308
_ex_1, 302
_ex_10, 300

_ex_11, 299
_ex_12, 299
_ex_120, 297
_ex_15, 299
_ex_18, 299
_ex_1_2, 302
_ex_1_3, 303
_ex_1_4, 303
_ex_2, 302
_ex_20, 298
_ex_24, 298
_ex_25, 298
_ex_3, 301
_ex_30, 298
_ex_4, 301
_ex_48, 297
_ex_5, 301
_ex_6, 301
_ex_60, 297
_ex_7, 300
_ex_8, 300
_ex_9, 300
_num0_bp, 295
_num0_p, 303
_num10_p, 308
_num11_p, 308
_num120_p, 311
_num12_p, 309
_num15_p, 309
_num18_p, 309
_num1_2_p, 304
_num1_3_p, 304
_num1_4_p, 304
_num1_p, 305
_num20_p, 309
_num24_p, 310
_num25_p, 310
_num2_p, 306
_num30_p, 310
_num3_p, 306
_num48_p, 310
_num4_p, 306
_num5_p, 307
_num60_p, 311
_num6_p, 307
_num7_p, 307
_num8_p, 307
_num9_p, 308
_num_10_p, 299
_num_11_p, 299
_num_120_p, 297
_num_12_p, 299
_num_15_p, 299
_num_18_p, 298
_num_1_2_p, 302
_num_1_3_p, 302
_num_1_4_p, 303
_num_1_p, 302
_num_20_p, 298
_num_24_p, 298
_num_25_p, 298
_num_2_p, 301
_num_30_p, 297
_num_3_p, 301
_num_48_p, 297
_num_4_p, 301
_num_5_p, 301
_num_60_p, 297
_num_6_p, 300
_num_7_p, 300
_num_8_p, 300
_num_9_p, 300
abs, 248
abs_conjugate, 150
abs_eval, 149
abs_evalf, 149
abs_expand, 149
abs_expl_derivative, 150
abs_imag_part, 151
abs_info, 151
abs_power, 151
abs_print_csrc_float, 150
abs_print_latex, 150
abs_real_part, 150
acos, 239
acos_conjugate, 193
acos_deriv, 193
acos_eval, 192
acos_evalf, 192
acosh, 242
acosh_conjugate, 202
acosh_deriv, 202
acosh_eval, 201
acosh_evalf, 201
adaptivesimpson, 204
add_symbol, 220
algebraic_match_mul_with_mul, 218
antisymmetric2, 276
antisymmetric3, 276
antisymmetric4, 276
antisymmetrize, 126, 277, 281
archive_atom, 68
archive_node_id, 68
are_ex_trivially_equal, 119
asin, 239
asin_conjugate, 192
asin_deriv, 191
asin_eval, 191
asin_evalf, 191
asin_info, 192
asinh, 242
asinh_conjugate, 201
asinh_deriv, 201
asinh_eval, 200
asinh_evalf, 200
atan, 240

atan2_deriv, 195
atan2_eval, 195
atan2_evalf, 195
atan2_info, 195
atan_conjugate, 194
atan_deriv, 194
atan_eval, 193
atan_evalf, 193
atan_info, 194
atan_series, 194
atanh, 242
atanh_conjugate, 203
atanh_deriv, 203
atanh_eval, 202
atanh_evalf, 202
atanh_series, 203
base_and_index, 100
bernoulli, 247
Bernoulli_polynomial, 206
beta_deriv, 173
beta_eval, 173
beta_evalf, 173
beta_series, 173
binomial, 246
binomial_conjugate, 160
binomial_eval, 160
binomial_evalf, 159
binomial_imag_part, 160
binomial_real_part, 160
binomial_sym, 160
callback_registered, 92
canonicalize, 276
canonicalize_clifford, 105
Catalan, 294
CatalanEvalf, 252
charpoly, 216
class_info< OPT >::first, 293
class_info< OPT >::parents_identified, 293
clifford_bar, 112
clifford_inverse, 107
clifford_max_label, 106
clifford_moebius_map, 109, 110
clifford_norm, 107
clifford_prime, 106
clifford_star, 112
clifford_star_bar, 105
clifford_to_lst, 109
clifford_unit, 101
coeff, 122
coerce, 236
coerce< int, cln::cl_I >, 236
coerce< unsigned int, cln::cl_I >, 236
collect, 123
collect_common_factors, 233
collect_symbols, 220
color_d, 115
color_f, 115
color_h, 116
color_ONE, 114
color_T, 114
color_trace, 117
cols, 215
compare_pointers, 287
compile_ex, 130, 131
conjugate, 120
conjugate_conjugate, 144
conjugate_eval, 144
conjugate_evalf, 143
conjugate_expl_derivative, 144
conjugate_funcp, 70
conjugate_funcp_1, 71
conjugate_funcp_10, 83
conjugate_funcp_11, 84
conjugate_funcp_12, 85
conjugate_funcp_13, 87
conjugate_funcp_14, 88
conjugate_funcp_2, 72
conjugate_funcp_3, 74
conjugate_funcp_4, 75
conjugate_funcp_5, 76
conjugate_funcp_6, 77
conjugate_funcp_7, 79
conjugate_funcp_8, 80
conjugate_funcp_9, 81
conjugate_funcp_exvector, 90
conjugate_imag_part, 145
conjugate_info, 145
conjugate_print_latex, 144
conjugate_real_part, 144
conjugateepvector, 133
convert_H_to_Li, 166
cos, 238
cos_conjugate, 189
cos_deriv, 188
cos_eval, 188
cos_evalf, 188
cos_imag_part, 189
cos_real_part, 189
cosh, 241
cosh_conjugate, 198
cosh_deriv, 198
cosh_eval, 197
cosh_evalf, 197
cosh_imag_part, 198
cosh_real_part, 198
count_dummy_indices, 138
count_free_indices, 138
crc32, 119
crctab, 294
csgn, 254
csgn_conjugate, 153
csgn_eval, 153
csgn_evalf, 153
csgn_imag_part, 154
csgn_power, 154
csgn_real_part, 154

csgn_series, 153
csrc, 267
csrc_cl_N, 268
csrc_double, 268
csrc_float, 268
cyclic_permutation, 289
decomp_rational, 223
degree, 121
delta_tensor, 283
denom, 122, 258
derivative_funcp, 70
derivative_funcp_1, 72
derivative_funcp_10, 83
derivative_funcp_11, 84
derivative_funcp_12, 86
derivative_funcp_13, 87
derivative_funcp_14, 89
derivative_funcp_2, 73
derivative_funcp_3, 74
derivative_funcp_4, 75
derivative_funcp_5, 76
derivative_funcp_6, 78
derivative_funcp_7, 79
derivative_funcp_8, 80
derivative_funcp_9, 82
derivative_funcp_exvector, 90
determinant, 215
dflt, 267
diag_matrix, 213
diff, 124
Digits, 296
digits_changed_callback, 92
dirac_gamma, 101
dirac_gamma5, 101
dirac_gammaL, 102
dirac_gammaR, 102
dirac_ONE, 100
dirac_slash, 103
dirac_trace, 104, 105
dirichlet_character, 206
divide, 224
divide_in_z, 225
doublefactorial, 246
dynallocate, 97, 98
EllipticE_deriv, 168
EllipticE_eval, 168
EllipticE_evalf, 167
EllipticE_print_latex, 168
EllipticE_series, 168
EllipticK_deriv, 167
EllipticK_eval, 166
EllipticK_evalf, 166
EllipticK_print_latex, 167
EllipticK_series, 167
epp, 69
epsilon_tensor, 284, 285
epvector, 69
eta_conjugate, 155
eta_eval, 155
eta_evalf, 154
eta_imag_part, 155
eta_real_part, 155
eta_series, 155
Euler, 294
EulerEvalf, 252
eval, 123
eval_funcp, 70
eval_funcp_1, 71
eval_funcp_10, 82
eval_funcp_11, 84
eval_funcp_12, 85
eval_funcp_13, 87
eval_funcp_14, 88
eval_funcp_2, 72
eval_funcp_3, 73
eval_funcp_4, 75
eval_funcp_5, 76
eval_funcp_6, 77
eval_funcp_7, 78
eval_funcp_8, 80
eval_funcp_9, 81
eval_funcp_exvector, 90
eval_integ, 124
evalf, 124, 215
evalf_funcp, 70
evalf_funcp_1, 71
evalf_funcp_10, 83
evalf_funcp_11, 84
evalf_funcp_12, 85
evalf_funcp_13, 87
evalf_funcp_14, 88
evalf_funcp_2, 72
evalf_funcp_3, 73
evalf_funcp_4, 75
evalf_funcp_5, 76
evalf_funcp_6, 77
evalf_funcp_7, 79
evalf_funcp_8, 80
evalf_funcp_9, 81
evalf_funcp_exvector, 90
evalffunctype, 69
evalm, 124
ex_to, 128
exadd, 258
exhashmap, 91
exmap, 69
exminus, 258
exmul, 258
exp, 237
exp_conjugate, 184
exp_deriv, 183
exp_eval, 183
exp_evalf, 182
exp_expand, 183
exp_imag_part, 183
exp_info, 184

exp_power, 184
exp_real_part, 183
expand, 120, 215
expand_dummy_sum, 143
expand_funcp, 70
expand_funcp_1, 71
expand_funcp_10, 83
expand_funcp_11, 84
expand_funcp_12, 86
expand_funcp_13, 87
expand_funcp_14, 89
expand_funcp_2, 73
expand_funcp_3, 74
expand_funcp_4, 75
expand_funcp_5, 76
expand_funcp_6, 78
expand_funcp_7, 79
expand_funcp_8, 80
expand_funcp_9, 82
expand_funcp_exvector, 90
expl_derivative_funcp, 70
expl_derivative_funcp_1, 72
expl_derivative_funcp_10, 83
expl_derivative_funcp_11, 85
expl_derivative_funcp_12, 86
expl_derivative_funcp_13, 88
expl_derivative_funcp_14, 89
expl_derivative_funcp_2, 73
expl_derivative_funcp_3, 74
expl_derivative_funcp_4, 75
expl_derivative_funcp_5, 77
expl_derivative_funcp_6, 78
expl_derivative_funcp_7, 79
expl_derivative_funcp_8, 81
expl_derivative_funcp_9, 82
expl_derivative_funcp_exvector, 90
exprseq, 70
exset, 69
exvector, 69
exvectorvector, 91
factor, 133
factorial, 245
factorial_conjugate, 159
factorial_eval, 158
factorial_evalf, 158
factorial_imag_part, 159
factorial_print_dflt_latex, 158
factorial_real_part, 159
fibonacci, 247
find, 121
find_common_factor, 233
find_dummy_indices, 138
find_factory_fcn, 95
find_free_and_dummy, 136, 138
find_variant_indices, 140
force_include_tgamma, 295
force_include_zeta1, 295
format_index_value, 289
frac_cancel, 233
fsolve, 163
func_arg_info, 145
FUNCP_1P, 69
FUNCP_2P, 69
FUNCP_CUBA, 69
G, 164
G2_eval, 176
G2_evalf, 175
G3_eval, 176
G3_evalf, 176
gcd, 229, 251
gcd_pf_mul, 228
gcd_pf_pow, 228
gcd_pf_pow_pow, 229
generalised_Bernoulli_number, 206
get_all_dummy_indices, 142
get_all_dummy_indices_safely, 141
get_clifford_comp, 108
get_default_TeX_name, 273
get_dim_uint, 100
get_first_symbol, 219
get_print_context, 265
get_print_options, 266
get_representation_label, 103, 116
get_symbol_stats, 220
GINAC_BIND_UNARCHIVER, 93, 99, 113, 119, 134–136, 139, 204, 207–210, 212, 218, 219, 235, 269, 271–274, 282, 292, 295
GINAC_DECLARE_UNARCHIVER, 93, 111, 112, 118, 119, 134, 135, 137, 143, 204, 210–212, 214, 218, 219, 253, 269, 271, 273, 274, 278, 286, 287, 292
GINAC_IMPLEMENT_REGISTERED_CLASS_OPT, 93, 95, 98, 112, 118, 132–135, 139, 203, 207–210, 212, 217, 218, 234, 269, 270, 272–274, 281, 292
golden_ratio_hash, 288
guess_precision, 244
H_deriv, 180
H_eval, 180
H_evalf, 179
H_print_latex, 180
H_series, 180
has, 121
hasindex, 141
haswild, 292
heur_gcd, 227
heur_gcd_z, 226
hold_ncmul, 219
I, 295
idx, 295
idx_symmetrization, 141
ifactor, 205
imag, 257
imag_part, 121
imag_part_conjugate, 148
imag_part_eval, 148

imag_part_evalf, 147
imag_part_expl_derivative, 149
imag_part_funcp, 70
imag_part_funcp_1, 71
imag_part_funcp_10, 83
imag_part_funcp_11, 84
imag_part_funcp_12, 86
imag_part_funcp_13, 87
imag_part_funcp_14, 89
imag_part_funcp_2, 73
imag_part_funcp_3, 74
imag_part_funcp_4, 75
imag_part_funcp_5, 76
imag_part_funcp_6, 78
imag_part_funcp_7, 79
imag_part_funcp_8, 80
imag_part_funcp_9, 82
imag_part_funcp_exvector, 90
imag_part_imag_part, 148
imag_part_print_latex, 148
imag_part_real_part, 148
index0, 275
index1, 275
index2, 275
index3, 275
index_dimensions, 268
indices_consistent, 139
info_funcp, 71
info_funcp_1, 72
info_funcp_10, 84
info_funcp_11, 85
info_funcp_12, 87
info_funcp_13, 88
info_funcp_14, 90
info_funcp_2, 73
info_funcp_3, 74
info_funcp_4, 76
info_funcp_5, 77
info_funcp_6, 78
info_funcp_7, 80
info_funcp_8, 81
info_funcp_9, 82
info_funcp_exvector, 91
interpolate, 226
inverse, 216, 253
iquo, 250
irem, 249
is_a, 95, 128, 270
is_cinteger, 256
is_clifford_tinfo, 112
is_color_tinfo, 116
is_crational, 256
is_dirac_slash, 100
is_discriminant_of_quadratic_number_field, 205
is_dummy_pair, 136
is_even, 255
is_exactly_a, 96, 128
is_integer, 254
is_negative, 254
is_nonneg_integer, 255
is_odd, 255
is_order_function, 166
is_polynomial, 121
is_pos_integer, 255
is_positive, 254
is_prime, 255
is_rational, 256
is_real, 256
is_terminating, 271
is_the_function, 135
is_the_function< G_SERIAL >, 164
is_the_function< iterated_integral_SERIAL >, 166
is_the_function< psi_SERIAL >, 165
is_the_function< zeta_SERIAL >, 164
is_zero, 127, 254
isqrt, 252
iterated_integral, 165
iterated_integral2_eval, 169
iterated_integral2_evalf, 169
iterated_integral3_eval, 169
iterated_integral3_evalf, 169
iterated_integral_evalf_impl, 169
kronecker_symbol, 205
latex, 267
lcm, 230, 251
lcm_of_coefficients_denominators, 221
lcmcoeff, 221
ldegree, 122
lgamma, 244
lgamma_conjugate, 171
lgamma_deriv, 170
lgamma_eval, 170
lgamma_evalf, 170
lgamma_series, 170
lhs, 126
Li2, 243
Li2_, 243
Li2_conjugate, 157
Li2_deriv, 156
Li2_eval, 156
Li2_evalf, 156
Li2_projection, 243
Li2_series, 156, 243
Li3_eval, 157
Li_deriv, 177
Li_eval, 177
Li_evalf, 176
Li_print_latex, 177
Li_series, 177
library_initializer, 294
link_ex, 131, 132
log, 237
log2, 287
log_conjugate, 186
log_deriv, 185
log_eval, 185

log_evalf, 184
log_expand, 186
log_imag_part, 185
log_info, 186
log_real_part, 185
log_series, 185
lookup_map, 91
lorentz_eps, 285
lorentz_g, 283
lsolve, 162
lst, 91
lst_to_clifford, 107, 108
lst_to_matrix, 213
make_real_float, 234
make_return_type_t, 272
map_eval_integ, 293
map_evalm, 293
match, 125
metric_tensor, 283
minimal_dim, 137
mod, 248
multinomial_coefficient, 287
multiply_lcm, 221
my_ios_callback, 265
my_ios_index, 265
next_print_context_id, 296
no_index_dimensions, 268
nops, 120, 214
normal, 123
not_symmetric, 275
number_of_type, 139
numer, 122, 257
numer_denom, 122
op, 126
operator!=, 264
operator<, 264
operator<<, 94, 119, 120, 253, 266, 289–291
operator<=, 264
operator>, 265
operator>>, 94, 95, 266
operator>=, 265
operator+, 258, 259, 261, 262
operator++, 262, 263
operator+=, 260, 261
operator-, 259, 261, 262
operator--, 262–264
operator-=, 260, 261
operator/, 259, 260
operator/=, 260, 261
operator==, 264
operator*, 259, 260
operator*=, 260, 261
Order_conjugate, 161
Order_eval, 161
Order_expl_derivative, 162
Order_imag_part, 162
Order_power, 162
Order_real_part, 161
Order_series, 161
paramset, 70
permutation_sign, 288
permute_free_index_to_front, 114
Pi, 293
PiEvalf, 252
pow, 253, 269, 270
power_funcp, 71
power_funcp_1, 72
power_funcp_10, 83
power_funcp_11, 85
power_funcp_12, 86
power_funcp_13, 88
power_funcp_14, 89
power_funcp_2, 73
power_funcp_3, 74
power_funcp_4, 75
power_funcp_5, 77
power_funcp_6, 78
power_funcp_7, 79
power_funcp_8, 81
power_funcp_9, 82
power_funcp_exvector, 90
prem, 223
primitive_dirichlet_character, 205
print_context_class_info, 92
print_func< print_context >, 135
print_func< print_dflt >, 98, 113, 282
print_funcp, 71
print_funcp_1, 72
print_funcp_10, 84
print_funcp_11, 85
print_funcp_12, 86
print_funcp_13, 88
print_funcp_14, 89
print_funcp_2, 73
print_funcp_3, 74
print_funcp_4, 76
print_funcp_5, 77
print_funcp_6, 78
print_funcp_7, 80
print_funcp_8, 81
print_funcp_9, 82
print_funcp_exvector, 91
print_integer_csrc, 235
print_operator, 273
print_real_cl_N, 237
print_real_csrc, 236
print_real_number, 235
print_sym_pow, 269
product_to_exvector, 140
psi, 165, 245
psi1_deriv, 174
psi1_eval, 174
psi1_evalf, 174
psi1_series, 174
psi2_deriv, 175
psi2_eval, 175

psi2_evalf, 174
psi2_series, 175
python, 267
python_repr, 267
quo, 222
rank, 216, 217
read_real_float, 235
read_unsigned, 94
real, 257
real_part, 120
real_part_conjugate, 146
real_part_eval, 146
real_part_evalf, 146
real_part_expl_derivative, 147
real_part_funcp, 70
real_part_funcp_1, 71
real_part_funcp_10, 83
real_part_funcp_11, 84
real_part_funcp_12, 86
real_part_funcp_13, 87
real_part_funcp_14, 89
real_part_funcp_2, 72
real_part_funcp_3, 74
real_part_funcp_4, 75
real_part_funcp_5, 76
real_part_funcp_6, 77
real_part_funcp_7, 79
real_part_funcp_8, 80
real_part_funcp_9, 81
real_part_funcp_exvector, 90
real_part_imag_part, 147
real_part_print_latex, 146
real_part_real_part, 147
reduced_matrix, 214
reeval_ncmul, 219
REGISTER_FUNCTION, 145, 147, 149, 151, 153, 154, 156–159, 161, 162, 167, 168, 171–173, 178, 179, 181, 184, 186, 188, 189, 191–193, 195–198, 200–203
registered_class_info, 92
rem, 222
remove_dirac_ONE, 106
rename_dummy_indices, 139
rename_dummy_indices_uniquely, 142, 143
replace_with_symbol, 232
reposition_dummy_indices, 140
resultant, 234
rhs, 127
rotate_left, 287
rows, 215
S_deriv, 179
S_eval, 178
S_evalf, 178
S_print_latex, 179
S_series, 178
series, 124
series_funcp, 71
series_funcp_1, 72
series_funcp_10, 83
series_funcp_11, 85
series_funcp_12, 86
series_funcp_13, 88
series_funcp_14, 89
series_funcp_2, 73
series_funcp_3, 74
series_funcp_4, 75
series_funcp_5, 77
series_funcp_6, 78
series_funcp_7, 79
series_funcp_8, 81
series_funcp_9, 82
series_funcp_exvector, 91
series_to_poly, 271
set_print_context, 266
set_print_func, 272
set_print_options, 266
shaker_sort, 288
simplify_indexed, 125, 141
simplify_indexed_product, 141
sin, 238
sin_conjugate, 187
sin_deriv, 187
sin_eval, 187
sin_evalf, 186
sin_imag_part, 187
sin_real_part, 187
sinh, 241
sinh_conjugate, 197
sinh_deriv, 196
sinh_eval, 196
sinh_evalf, 196
sinh_imag_part, 197
sinh_real_part, 196
smod, 248
spinor_metric, 284
spmap, 91
sprem, 224
sqrfree, 231
sqrfree_parfrac, 231
sqrfree_yun, 230
sqrt, 251, 270
sr_gcd, 226
step, 254
step_conjugate, 152
step_eval, 152
step_evalf, 151
step_imag_part, 152
step_real_part, 152
step_series, 152
structure< T, CP >::reg_info, 296
sub_matrix, 214
subs, 127, 128
subsvalue, 204
swap, 127, 132
sy_anti, 279, 280
sy_cycl, 280, 281

sy_none, 278
sy_symm, 279
sym_desc_vec, 92
symbolic_matrix, 213, 217
symm, 277
symmetric2, 275
symmetric3, 276
symmetric4, 276
symmetrize, 125, 277, 281
symmetrize_cyclic, 126, 277, 281
synthesize_func, 68
tan, 239
tan_conjugate, 191
tan_deriv, 190
tan_eval, 190
tan_evalf, 189
tan_imag_part, 190
tan_real_part, 190
tan_series, 190
tanh, 241
tanh_conjugate, 200
tanh_deriv, 199
tanh_eval, 199
tanh_evalf, 199
tanh_imag_part, 200
tanh_real_part, 199
tanh_series, 199
tensor, 293
tgamma, 244, 245
tgamma_conjugate, 172
tgamma_deriv, 172
tgamma_eval, 171
tgamma_evalf, 171
tgamma_series, 172
to_double, 257
to_int, 256
to_long, 257
to_polynomial, 123
to_rational, 123
trace, 216
trace_string, 103
transpose, 215
tree, 267
trig_info, 188
tryfactsubs, 217
uintvector, 91
unarch_table_instance, 293
unarchive_map_t, 68
unit_matrix, 213, 217
unlink_ex, 132
unsignedvector, 91
version_major, 296
version_micro, 297
version_minor, 296
wild, 292
write_real_float, 235
write_unsigned, 94
zeta, 163, 164, 244
  zeta1_deriv, 181
  zeta1_eval, 181
  zeta1_evalf, 181
  zeta1_print_latex, 181
  zeta2_deriv, 182
  zeta2_eval, 182
  zeta2_evalf, 182
  zeta2_print_latex, 182
  zetaderiv_deriv, 158
  zetaderiv_eval, 157
ginac Directory Reference, 23
ginac.h, 1306
GiNaC::_numeric_digits, 317
  _numeric_digits, 318
  add_callback, 318
  callbacklist, 319
  digits, 319
  operator long, 318
  operator=, 318
  print, 318
  too_late, 319
GiNaC::add, 320
  add, 326, 327
  coeff, 328
  combine_ex_with_coeff_to_pair, 334
  combine_pair_with_coeff_to_pair, 334
  conjugate, 331
  degree, 328
  derivative, 332
  do_print, 335
  do_print_csrc, 335
  do_print_latex, 335
  do_print_python_repr, 336
  eval, 329
  eval_ncmul, 332
  evalm, 329
  expand, 334
  get_free_indices, 332
  imag_part, 332
  info, 327
  integer_content, 330
  is_polynomial, 328
  ldegree, 328
  max_coefficient, 331
  mul, 336
  normal, 330
  power, 336
  precedence, 327
  print_add, 335
  real_part, 331
  recombine_pair_to_ex, 334
  return_type, 332
  return_type_tinfo, 333
  series, 329
  smod, 330
  split_ex_to_pair, 333
  thisexpairseq, 333
GiNaC::archive, 337

~archive, 338
add_node, 341
archive, 338, 339
archive_ex, 339
atomize, 342
atoms, 343
clear, 341
exprs, 343
exprtable, 344
forget, 341
get_node, 341
get_top_node, 341
inv_at_cit, 338
inverse_atoms, 344
nodes, 343
num_expressions, 340
operator<<, 343
operator>>, 343
printraw, 342
unarchive_ex, 339, 340
unatomize, 342
GiNaC::archive::archived_ex, 357
archived_ex, 357
name, 358
root, 358
GiNaC::archive_node, 344
a, 354
add_bool, 348
add_ex, 349
add_string, 349
add_unsigned, 349
archive_node, 348
archive_node_cit, 347
e, 355
find_bool, 349
find_ex, 351
find_ex_by_loc, 351
find_ex_node, 352
find_first, 350
find_last, 351
find_property_range, 351
find_string, 350
find_unsigned, 350
forget, 353
get_ex, 353
get_properties, 352
has_ex, 353
has_expression, 354
has_same_ex_as, 352
operator<<, 354
operator>>, 354
operator=, 348
printraw, 353
property_type, 348
propinfovector, 347
props, 354
PTYPE_BOOL, 348
PTYPE_NODE, 348
PTYPE_STRING, 348
PTYPE_UNSIGNED, 348
unarchive, 352
GiNaC::archive_node::archive_node_cit_range, 355
begin, 356
end, 356
GiNaC::archive_node::property, 1043
name, 1044
property, 1043
type, 1044
value, 1044
GiNaC::archive_node::property_info, 1044
count, 1045
name, 1045
property_info, 1045
type, 1045
GiNaC::basic, 358
~basic, 363
accept, 370
add_indexed, 376
archive, 380
basic, 363
calchash, 379
clearflag, 384
coeff, 371
collect, 372
compare, 382
compare_same_type, 379
conjugate, 378
contract_with, 377
dbgprint, 366
dbgprinttree, 366
degree, 371
derivative, 372
diff, 382
do_print, 385
do_print_python_repr, 385
do_print_tree, 385
duplicate, 364
ensure_if_modifiable, 384
eval, 364
eval_indexed, 365
eval_integ, 365
eval_ncmul, 365
evalf, 364
evalm, 364
ex, 386
expand, 372
flags, 387
get_free_indices, 375
gethash, 383
has, 369
hashvalue, 387
hold, 383
imag_part, 378
info, 367
integer_content, 374
is_equal, 382

is_equal_same_type, 379
is_polynomial, 371
ldegree, 371
let_op, 368
map, 370
match, 369
match_same_type, 369
max_coefficient, 375
nops, 367
normal, 373
op, 367
operator=, 364
operator[], 368, 369
precedence, 366
print, 365
print_dispatch, 380
read_archive, 381
real_part, 378
return_type, 377
return_type_tinfo, 378
scalar_mul_indexed, 376
series, 373
setflag, 384
smod, 375
subs, 370
subs_one_level, 381
to_polynomial, 374
to_rational, 374
GiNaC::basic_log_kernel, 388
do_print, 393
series_coeff_impl, 393
GiNaC::basic_multi_iterator< T >, 394
~basic_multi_iterator, 396
B, 399
basic_multi_iterator, 396
flag_overflow, 400
get_vector, 397
init, 398
N, 399
operator<<, 399
operator(), 398
operator++, 398
operator[], 397, 398
overflow, 397
size, 397
v, 399
GiNaC::basic_partition_generator, 400
basic_partition_generator, 402
mpgen, 402
GiNaC::basic_partition_generator::mpartition2, 852
m, 853
mpartition2, 852
n, 853
next_partition, 852
x, 853
GiNaC::class_info< OPT >, 402
class_info, 404
dump_hierarchy, 404
dump_tree, 404
find, 404
first, 405
get_parent, 404
identify_parents, 405
next, 405
options, 405
parent, 405
parents_identified, 405
GiNaC::class_info< OPT >::tree_node, 1232
add_child, 1233
children, 1233
info, 1233
tree_node, 1233
GiNaC::clifford, 406
archive, 414
clifford, 413, 414
commutator_sign, 419
do_print_dflt, 418
do_print_latex, 419
do_print_tree, 419
eval_ncmul, 415
get_commutator_sign, 417
get_metric, 417
get_representation_label, 417
let_op, 418
match_same_type, 415
metric, 419
nops, 417
op, 418
precedence, 414
read_archive, 415
representation_label, 419
return_type, 416
return_type_tinfo, 416
same_metric, 417
subs, 418
thiscontainer, 416
GiNaC::cliffordunit, 420
contract_with, 425
do_print, 425
do_print_latex, 425
GiNaC::color, 425
archive, 434
color, 433, 434
eval_ncmul, 434
get_representation_label, 436
match_same_type, 435
read_archive, 434
representation_label, 436
return_type, 436
return_type_tinfo, 436
thiscontainer, 435
GiNaC::compare_all_equal< T >, 437
~compare_all_equal, 437
struct_compare, 437
struct_is_equal, 437
GiNaC::compare_bitwise< T >, 438

~compare_bitwise, 438
struct_compare, 438
struct_is_equal, 438
GiNaC::compare_std_less< T >, 439
~compare_std_less, 439
struct_compare, 439
struct_is_equal, 439
GiNaC::composition_generator, 440
atend, 441
cmgen, 441
composition, 442
composition_generator, 441
current_updated, 442
get, 441
next, 441
trivial, 441
GiNaC::composition_generator::coolmulti, 485
~coolmulti, 485
after_i, 486
coolmulti, 485
finished, 486
head, 486
i, 486
next_permutation, 486
GiNaC::composition_generator::coolmulti::element, 545
~element, 546
element, 546
next, 546
value, 546
GiNaC::const_iterator, 442
const_iterator, 445
const_postorder_iterator, 448
const_preorder_iterator, 447
difference_type, 444
e, 448
ex, 447
i, 448
iterator_category, 444
operator!=, 447
operator<, 447
operator<=, 447
operator>, 447
operator>=, 447
operator+, 446, 448
operator++, 445
operator+=, 446
operator-, 446, 448
operator->, 445
operator--, 446
operator-=, 446
operator==, 446
operator[], 445
operator*, 445
pointer, 444
reference, 444
value_type, 444
GiNaC::const_postorder_iterator, 449
const_postorder_iterator, 450
descend, 451
difference_type, 449
increment, 451
iterator_category, 449
operator!=, 451
operator++, 450
operator->, 450
operator==, 451
operator*, 450
pointer, 449
reference, 450
s, 451
value_type, 449
GiNaC::const_preorder_iterator, 452
const_preorder_iterator, 453
difference_type, 452
increment, 454
iterator_category, 452
operator!=, 454
operator++, 453
operator->, 453
operator==, 454
operator*, 453
pointer, 452
reference, 453
s, 454
value_type, 452
GiNaC::constant, 455
archive, 462
calchash, 463
conjugate, 461
constant, 460
derivative, 462
do_print, 464
do_print_latex, 464
do_print_python_repr, 464
do_print_tree, 464
domain, 465
ef, 465
evalf, 461
get_name, 463
get_TeX_name, 463
imag_part, 462
info, 461
is_equal_same_type, 463
is_polynomial, 461
name, 465
next_serial, 465
number, 465
read_archive, 462
real_part, 461
serial, 465
TeX_name, 465
GiNaC::container< C >, 466
append, 478
archive, 475
begin, 479
conjugate, 475

const_iterator, 472
const_reverse_iterator, 472
container, 472
do_print, 480
do_print_python, 480
do_print_python_repr, 480
do_print_tree, 480
end, 479
get_close_delim, 473
get_default_flags, 473
get_open_delim, 473
imag_part, 476
info, 473
is_equal_same_type, 476
let_op, 474
nops, 473
op, 474
precedence, 473
prepend, 478
printseq, 477
rbegin, 480
read_archive, 475
real_part, 476
remove_all, 479
remove_first, 478
remove_last, 478
rend, 480
sort, 479
sort_, 477
STLT, 472
subs, 474
subschildren, 481
thiscontainer, 476, 477
unique, 479
unique_, 478, 481
GiNaC::container_storage< C >, 481
~container_storage, 483
container_storage, 482, 483
reserve, 483
seq, 484
STLT, 482
GiNaC::derivative_map_function, 487
derivative_map_function, 488
operator(), 488
s, 488
GiNaC::determinant_algo, 489
automatic, 489
bareiss, 490
divfree, 489
gauss, 489
laplace, 489
GiNaC::diracgamma, 490
contract_with, 495
do_print, 495
do_print_latex, 495
GiNaC::diracgamma5, 496
conjugate, 500
do_print, 500
do_print_latex, 500
GiNaC::diracgammaL, 501
conjugate, 505
do_print, 505
do_print_latex, 505
GiNaC::diracgammaR, 506
conjugate, 510
do_print, 510
do_print_latex, 510
GiNaC::diracone, 511
do_print, 515
do_print_latex, 515
GiNaC::do_taylor, 516
GiNaC::domain, 516
complex, 516
positive, 516
real, 516
GiNaC::dunno, 517
GiNaC::Ebar_kernel, 517
do_print, 524
Ebar_kernel, 523
get_numerical_value, 524
is_numeric, 523
let_op, 523
m, 525
n, 525
nops, 523
op, 523
series_coeff_impl, 524
x, 525
y, 525
GiNaC::Eisenstein_h_kernel, 525
C_norm, 535
coefficient_a0, 534
coefficient_an, 534
do_print, 534
Eisenstein_h_kernel, 532
get_numerical_value, 533
is_numeric, 533
k, 535
Laurent_series, 533
let_op, 532
N, 535
nops, 532
op, 532
q_expansion_modular_form, 534
r, 535
s, 535
series, 532
uses_Laurent_series, 533
GiNaC::Eisenstein_kernel, 536
a, 545
b, 545
C_norm, 545
do_print, 544
Eisenstein_kernel, 542
get_numerical_value, 543
is_numeric, 543

K, 545
k, 544
Laurent_series, 543
let_op, 543
N, 544
nops, 542
op, 543
q_expansion_modular_form, 544
series, 542
uses_Laurent_series, 544
GiNaC::ELi_kernel, 547
do_print, 554
ELi_kernel, 553
get_numerical_value, 554
is_numeric, 553
let_op, 553
m, 554
n, 554
nops, 553
op, 553
series_coeff_impl, 554
x, 555
y, 555
GiNaC::error_and_integral, 556
error, 557
error_and_integral, 557
integral, 557
GiNaC::error_and_integral_is_less, 557
operator(), 557
GiNaC::eval_integ_map_function, 558
operator(), 559
GiNaC::evalf_map_function, 559
operator(), 560
GiNaC::evalm_map_function, 561
operator(), 562
GiNaC::ex, 562
accept, 576
antisymmetrize, 589
archive_node, 593
are_ex_trivially_equal, 593
begin, 568
bp, 595
coeff, 578
collect, 579
compare, 587
conjugate, 573
construct_from_basic, 590
construct_from_double, 592
construct_from_int, 591
construct_from_long, 591
construct_from_longlong, 592
construct_from_string_and_lst, 592
construct_from_uint, 591
construct_from_ulong, 591
construct_from_ulonglong, 592
content, 583
dbgprint, 570
dbgprinttree, 571
degree, 577
denom, 581
diff, 579
end, 568
eval, 569
eval_integ, 570
eval_ncmul, 570
evalf, 569
evalm, 569
ex, 566–568
ex_to, 593
expand, 578
find, 574
get_free_indices, 586
gethash, 590
has, 574
imag_part, 574
info, 571
integer_content, 583
is_a, 594
is_equal, 587
is_exactly_a, 594
is_polynomial, 577
is_zero, 588
is_zero_matrix, 588
lcoeff, 578
ldegree, 577
let_op, 573
lhs, 573
makewriteable, 592
map, 576
match, 575
max_coefficient, 586
nops, 571
normal, 580
numer, 581
numer_denom, 582
op, 572
operator[], 572, 573
postorder_begin, 569
postorder_end, 569
preorder_begin, 568
preorder_end, 569
primpart, 584
print, 570
real_part, 574
return_type, 590
return_type_tinfo, 590
rhs, 573
series, 579
share, 593
simplify_indexed, 586, 587
smod, 585
subs, 575, 576
swap, 568
symmetrize, 589
symmetrize_cyclic, 589, 590
tcoeff, 578

to_polynomial, 581
to_rational, 580
traverse, 577
traverse_postorder, 577
traverse_preorder, 576
unit, 582
unitcontprim, 585
GiNaC::ex_base_is_less, 595
operator(), 595
GiNaC::ex_is_equal, 595
operator(), 596
GiNaC::ex_is_less, 596
operator(), 596
GiNaC::ex_swap, 596
operator(), 597
GiNaC::expair, 597
coeff, 601
compare, 600
conjugate, 601
expair, 599
is_canonical_numeric, 600
is_equal, 600
is_less, 600
print, 600
rest, 601
swap, 600
GiNaC::expair_is_less, 602
operator(), 602
GiNaC::expair_rest_is_less, 602
operator(), 603
GiNaC::expair_swap, 603
operator(), 603
GiNaC::expairseq, 604
archive, 613
calchash, 614
can_make_flat, 617
canonicalize, 619
combine_ex_with_coeff_to_pair, 616
combine_overall_coeff, 617
combine_pair_with_coeff_to_pair, 616
combine_same_terms_sorted_seq, 620
conjugate, 612
construct_from_2_ex, 618
construct_from_2_expairseq, 618
construct_from_epvector, 619
construct_from_expairseq_ex, 618
construct_from_exvector, 618
default_overall_coeff, 617
do_print, 617
do_print_tree, 618
eval, 611
evalchildren, 620
expair_needs_further_processing, 616
expairseq, 609, 610
expand, 614
expandchildren, 620
info, 610
is_canonical, 620
is_equal_same_type, 613
make_flat, 619
map, 611
match, 612
nops, 610
op, 611
overall_coeff, 621
precedence, 610
printpair, 615
printseq, 615
read_archive, 613
recombine_pair_to_ex, 616
return_type, 614
seq, 621
split_ex_to_pair, 615
subs, 612
subschildren, 621
thisexpairseq, 614, 615
to_polynomial, 612
to_rational, 611
GiNaC::expand_map_function, 622
expand_map_function, 623
operator(), 624
options, 624
GiNaC::expand_options, 624
expand_function_args, 625
expand_indexed, 625
expand_rename_idx, 625
expand_transcendental, 625
GiNaC::factor_options, 625
all, 625
polynomial, 625
GiNaC::fail, 626
do_print, 630
return_type, 630
GiNaC::fderivative, 630
archive, 640
derivative, 641
derivatives, 642
do_print, 642
do_print_csrc, 643
do_print_latex, 643
do_print_tree, 643
eval, 639
fderivative, 638, 639
is_equal_same_type, 641
match_same_type, 642
parameter_set, 644
print, 639
read_archive, 641
series, 640
thiscontainer, 640
GiNaC::function, 644
archive, 659
calchash, 657
conjugate, 659
current_serial, 664
derivative, 660

eval, 657
eval_ncmul, 657
evalf, 657
expand, 656
expl_derivative, 662
find_function, 663
function, 651–655
get_name, 663
get_registered_functions, 663
get_serial, 663
imag_part, 659
info, 660
is_equal_same_type, 660
lookup_remember_table, 662
match_same_type, 661
pderivative, 661
power, 662
precedence, 656
print, 656
read_archive, 660
real_part, 659
register_new, 663
registered_functions, 662
remember_table_entry, 664
return_type, 661
return_type_tinfo, 661
serial, 664
series, 658
store_remember_table, 662
thiscontainer, 658
GiNaC::function_options, 665
~function_options, 671
conjugate_f, 701
conjugate_func, 676–678, 695
conjugate_use_exvector_args, 704
derivative_f, 702
derivative_func, 684–686, 695
derivative_use_exvector_args, 704
do_not_evalf_params, 699
dummy, 672
eval_f, 701
eval_func, 672–674, 694
eval_use_exvector_args, 704
evalf_f, 701
evalf_func, 674–676, 694
evalf_params_first, 703
evalf_use_exvector_args, 704
expand_f, 702
expand_func, 682–684, 695
expand_use_exvector_args, 704
expl_derivative_f, 702
expl_derivative_func, 686–688, 695
expl_derivative_use_exvector_args, 704
fderivative, 700
function, 700
function_options, 671
functions_with_same_name, 705
get_name, 699
get_nparams, 699
has_derivative, 699
has_power, 700
imag_part_f, 702
imag_part_func, 680–682, 695
imag_part_use_exvector_args, 704
info_f, 702
info_func, 692–694, 696
info_use_exvector_args, 705
initialize, 672
latex_name, 672
name, 701
nparams, 701
overloaded, 699
power_f, 702
power_func, 688–690, 695
power_use_exvector_args, 705
print_dispatch_table, 702
print_func, 696–698
print_use_exvector_args, 705
real_part_f, 701
real_part_func, 678–680, 695
real_part_use_exvector_args, 704
remember, 699
remember_assoc_size, 703
remember_size, 703
remember_strategy, 703
return_type, 703
return_type_tinfo, 703
series_f, 702
series_func, 690–692, 696
series_use_exvector_args, 705
set_name, 672
set_print_func, 700
set_return_type, 698
set_symmetry, 699
symtree, 705
test_and_set_nparams, 700
TeX_name, 701
use_remember, 703
use_return_type, 703
GiNaC::G2_SERIAL, 706
serial, 706
GiNaC::G3_SERIAL, 706
serial, 707
GiNaC::gcd_options, 707
no_heur_gcd, 707
no_part_factored, 708
use_sr_gcd, 708
GiNaC::gcdheu_failed, 708
GiNaC::Gt, 708
apply_function_recursive, 721
applyIntegrationPath, 718
archive, 716
args, 723
auto_clear_polylog_cache, 723
calchash, 715
clear_polylog_cache, 721

cutIntegrationPath_threshold_imag, 723
cutIntegrationPath_threshold_real, 723
decomposeIntegrationPath, 717
do_print, 722
do_print_latex, 722
enable_tauToFundamental, 723
eval, 714
evalf, 715
evaluate, 718
ex_cutIntegrationPath, 719, 720
ex_evaluate, 719, 721
ex_prepare, 719, 721
ex_qExpand, 719, 721
ex_regularise, 718, 720
ex_tauToFundamental, 719, 720
ex_zisToFundamental, 718, 720
findMoebiusTransform, 717
getArgs, 714
getTau, 714
getZ, 714
Gt, 713
has, 716
lst_evaluate, 720, 721
match, 716
nargs, 723
nops, 715
nqexand_transformer, 724
precedence, 714
qExpand, 718
qExpand_minOrder, 723
qExpand_stepSize, 723
read_archive, 716
regularise, 717
setArgs, 714
subs, 715
tau, 724
tauToFundamental, 717
to_integer, 722
to_numeric, 722
z, 724
zisToFundamental, 717
GiNaC::Gt::kernel, 784
check, 786
deform, 787
Gt, 786
kernel, 786
ni, 787
zi, 787
GiNaC::Gt_detail, 311
deconcatenate_path, 312
deconcatenate_path< ex >, 312
deconcatenate_path< Gt::kernel >, 312
integer_partition, 312
operator<, 313
operator<<, 313
GiNaC::Gt_detail::pathintegral_term, 960
denom, 963
evaluate, 962
G_path, 962
integrate, 962
operator<, 963
operator<<, 962
pathintegral_term, 961
pathintegral_term_list, 962
polylog, 963
power, 963
prefactor, 963
GiNaC::Gt_detail::pathintegral_term_list, 964
add, 965
begin, 964
clear, 964
empty, 964
end, 964
pop, 965
size, 965
terms, 966
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1229
cache, 1232
check_type, 1231
clear_cache, 1231
func_default, 1232
func_obj, 1232
get_cache, 1231
impl, 1231
operator(), 1230
print_cache, 1230
TransformExpressionWithCache, 1230
GiNaC::has_distance< T >, 724
no_type, 725
test, 726
value, 726
yes_type, 725
GiNaC::has_options, 726
algebraic, 727
GiNaC::idx, 728
archive, 735
calchash, 736
derivative, 736
dim, 740
do_print, 739
do_print_csrc, 739
do_print_latex, 739
do_print_tree, 739
evalf, 734
get_dim, 738
get_value, 737
idx, 733
info, 734
is_dim_numeric, 738
is_dim_symbolic, 738
is_dummy_pair_same_type, 737
is_numeric, 737
is_symbolic, 737
map, 734
match_same_type, 736

minimal_dim, 738
nops, 734
op, 734
print_index, 739
read_archive, 735
replace_dim, 738
subs, 735
value, 740
GiNaC::idx_is_equal_ignore_dim, 740
operator(), 741
GiNaC::indexed, 741
all_index_values_are, 756
archive, 754
derivative, 754
do_print, 757
do_print_latex, 757
do_print_tree, 758
eval, 753
expand, 755
get_dummy_indices, 756
get_free_indices, 753
get_indices, 756
get_symmetry, 757
has_dummy_index_for, 756
imag_part, 753
indexed, 748–752
info, 752
precedence, 752
print_indexed, 757
printindices, 757
read_archive, 754
real_part, 753
reposition_dummy_indices, 759
return_type, 755
return_type_tinfo, 755
simplify_indexed, 758
simplify_indexed_product, 758
symtree, 759
thiscontainer, 754, 755
validate, 758
GiNaC::info_flags, 760
cinteger, 760
cinteger_polynomial, 761
crational, 760
crational_polynomial, 761
even, 760
expanded, 761
exprseq, 761
has_indices, 761
idx, 761
indefinite, 761
indexed, 761
integer, 760
integer_polynomial, 761
list, 761
negative, 760
negint, 760
nonnegative, 760
nonnegint, 760
numeric, 760
odd, 760
polynomial, 761
posint, 760
positive, 760
prime, 761
rational, 760
rational_function, 761
rational_polynomial, 761
real, 760
relation, 761
relation_equal, 761
relation_greater, 761
relation_greater_or_equal, 761
relation_less, 761
relation_less_or_equal, 761
relation_not_equal, 761
symbol, 761
GiNaC::integral, 761
a, 772
archive, 770
b, 772
conjugate, 770
degree, 768
derivative, 771
do_print, 771
do_print_latex, 772
eval, 767
eval_integ, 770
eval_ncmul, 768
evalf, 767
expand, 769
f, 773
get_free_indices, 769
integral, 767
ldegree, 768
let_op, 769
max_integration_level, 772
nops, 768
op, 768
precedence, 767
read_archive, 770
relative_integration_error, 772
return_type, 769
return_type_tinfo, 770
series, 771
x, 772
GiNaC::integration_kernel, 773
cache_step_size, 782
do_print, 781
get_cache_size, 780
get_numerical_value, 779
get_numerical_value_impl, 781
get_series_coeff, 780
has_trailing_zero, 778
is_numeric, 778
Laurent_series, 779

series, 778
series_coeff, 781
series_coeff_impl, 780
series_vec, 782
set_cache_step, 780
uses_Laurent_series, 779
GiNaC::internal, 313
GiNaC::internal::_iter_rep, 315
_iter_rep, 316
e, 316
i, 316
i_end, 317
operator!=, 316
operator==, 316
GiNaC::is_not_a_clifford, 782
operator(), 782
GiNaC::is_summation_idx, 783
operator(), 783
GiNaC::iterated_integral2_SERIAL, 783
serial, 783
GiNaC::iterated_integral3_SERIAL, 784
serial, 784
GiNaC::Kronecker_dtau_kernel, 787
C_norm, 795
do_print, 794
get_numerical_value, 794
is_numeric, 793
K, 795
Kronecker_dtau_kernel, 793
let_op, 793
n, 795
nops, 793
op, 793
series_coeff_impl, 794
z, 795
GiNaC::Kronecker_dz_kernel, 796
C_norm, 804
do_print, 803
get_numerical_value, 803
is_numeric, 802
K, 804
Kronecker_dz_kernel, 802
let_op, 802
n, 804
nops, 802
op, 802
series_coeff_impl, 803
tau, 804
z_j, 804
GiNaC::lanczos_coeffs, 804
calc_lanczos_A, 805
coeffs, 806
current_vector, 806
get_order, 805
lanczos_coeffs, 805
sufficiently_accurate, 805
GiNaC::library_init, 807
∼library_init, 808
count, 809
init_unarchivers, 808
library_init, 808
GiNaC::make_flat_inserter, 809
combine_indices, 810
do_renaming, 810
handle_factor, 810
make_flat_inserter, 810
used_indices, 810
GiNaC::map_function, 811
∼map_function, 813
argument_type, 813
operator(), 813
result_type, 813
GiNaC::matrix, 814
add, 825
add_indexed, 822
archive, 823
charpoly, 829
col, 835
cols, 824
conjugate, 823
contract_with, 822
determinant, 828
determinant_minor, 831
division_free_elimination, 832
do_print, 834
do_print_latex, 835
do_print_python_repr, 835
echelon_form, 832
eval_indexed, 822
evalm, 821
fraction_free_elimination, 833
gauss_elimination, 832
imag_part, 823
inverse, 829, 830
is_zero_matrix, 831
let_op, 821
m, 835
markowitz_elimination, 833
match_same_type, 824
matrix, 819, 820
mul, 825, 826
mul_scalar, 826
nops, 821
op, 821
operator(), 826, 827
pivot, 834
pow, 826
print_elements, 834
rank, 831
read_archive, 823
real_part, 823
return_type, 824
row, 835
rows, 824
scalar_mul_indexed, 822
set, 827

solve, 830
sub, 825
subs, 821
trace, 828
transpose, 827
GiNaC::minkmetric, 836
archive, 842
do_print, 842
do_print_latex, 842
eval_indexed, 841
info, 841
minkmetric, 841
pos_sig, 843
read_archive, 842
return_type, 842
GiNaC::modular_form_kernel, 843
C_norm, 851
do_print, 851
get_numerical_value, 850
is_numeric, 850
k, 851
Laurent_series, 850
let_op, 849
modular_form_kernel, 849
nops, 849
op, 849
P, 851
q_expansion_modular_form, 851
series, 849
uses_Laurent_series, 850
GiNaC::mul, 853
add, 871
algebraic_subs_mul, 869
can_be_further_expanded, 870
can_make_flat, 868
coeff, 861
combine_ex_with_coeff_to_pair, 867
combine_overall_coeff, 868
combine_pair_with_coeff_to_pair, 867
conjugate, 865
default_overall_coeff, 868
degree, 861
derivative, 865
do_print, 869
do_print_csrc, 870
do_print_latex, 870
do_print_python_repr, 870
eval, 862
eval_ncmul, 865
evalf, 862
evalm, 863
expair_needs_further_processing, 867
expand, 868
expandchildren, 870
find_real_imag, 869
get_free_indices, 865
has, 862
imag_part, 863
info, 860
integer_content, 864
is_polynomial, 861
ldegree, 861
max_coefficient, 864
mul, 859, 860
ncmul, 871
normal, 863
power, 871
precedence, 860
print_overall_coeff, 869
real_part, 863
recombine_pair_to_ex, 867
return_type, 866
return_type_tinfo, 866
series, 863
smod, 864
split_ex_to_pair, 866
thisexpairseq, 866
GiNaC::multi_iterator_counter< T >, 872
init, 874
multi_iterator_counter, 874
operator<<, 875
operator++, 874
GiNaC::multi_iterator_counter_indv< T >, 875
init, 878
multi_iterator_counter_indv, 877, 878
Nv, 879
operator<<, 879
operator++, 878
GiNaC::multi_iterator_ordered< T >, 879
init, 882
multi_iterator_ordered, 881, 882
operator<<, 883
operator++, 882
GiNaC::multi_iterator_ordered_eq< T >, 883
init, 886
multi_iterator_ordered_eq, 885
operator<<, 886
operator++, 886
GiNaC::multi_iterator_ordered_eq_indv< T >, 887
init, 889
multi_iterator_ordered_eq_indv, 889
Nv, 890
operator<<, 890
operator++, 889
GiNaC::multi_iterator_permutation< T >, 891
get_sign, 894
init, 894
multi_iterator_permutation, 893
operator<<, 895
operator++, 894
GiNaC::multi_iterator_shuffle< T >, 895
init, 898
multi_iterator_shuffle, 897
N_internal, 898
operator<<, 898
operator++, 898

v_internal, 898
v_orig, 899
GiNaC::multi_iterator_shuffle_prime< T >, 899
init, 902
multi_iterator_shuffle_prime, 902
operator<<, 902
GiNaC::multiple_polylog_kernel, 903
do_print, 910
is_numeric, 909
let_op, 909
multiple_polylog_kernel, 909
nops, 909
op, 909
series_coeff_impl, 909
z, 910
GiNaC::ncmul, 911
append_factors, 922
coeff, 919
conjugate, 920
count_factors, 922
degree, 918
derivative, 921
do_print, 922
do_print_csrc, 922
eval, 919
evalm, 919
expand, 918
expandchildren, 922
get_factors, 923
get_free_indices, 920
hold_ncmul, 923
imag_part, 921
info, 918
ldegree, 918
ncmul, 917
power, 923
precedence, 918
real_part, 921
reeval_ncmul, 923
return_type, 921
return_type_tinfo, 921
thiscontainer, 920
GiNaC::normal_map_function, 924
operator(), 925
GiNaC::numeric, 925
add, 940
add_dyn, 942
archive, 939
calchash, 940
coeff, 935
compare, 945
conjugate, 939
csgn, 945
degree, 935
denom, 952
derivative, 940
div, 941
div_dyn, 943
do_print, 952
do_print_csrc, 953
do_print_csrc_cl_N, 953
do_print_latex, 953
do_print_python_repr, 953
do_print_tree, 953
eval, 936
evalf, 936
has, 935
imag, 951
imag_part, 939
info, 934
int_length, 952
integer_content, 938
inverse, 944
is_cinteger, 948
is_crational, 948
is_equal, 945
is_equal_same_type, 940
is_even, 947
is_integer, 946
is_negative, 946
is_nonneg_integer, 947
is_odd, 947
is_polynomial, 934
is_pos_integer, 947
is_positive, 946
is_prime, 947
is_rational, 948
is_real, 948
is_zero, 946
ldegree, 935
max_coefficient, 938
mul, 941
mul_dyn, 942
normal, 937
numer, 951
numeric, 932, 933
operator!=, 949
operator<, 949
operator<=, 949
operator>, 949
operator>=, 950
operator=, 943, 944
operator==, 949
power, 942
power_dyn, 943
precedence, 934
print_numeric, 952
read_archive, 939
real, 951
real_part, 939
smod, 938
step, 944
sub, 941
sub_dyn, 942
subs, 936
to_cl_N, 951

to_double, 950
to_int, 950
to_long, 950
to_polynomial, 937
to_rational, 937
value, 954
GiNaC::op0_is_equal, 954
operator(), 954
GiNaC::partition_generator, 955
current_updated, 957
get, 956
next, 956
partition, 957
partition_generator, 956
GiNaC::partition_with_zero_parts_generator, 957
current_updated, 959
get, 959
m, 959
next, 959
partition, 959
partition_with_zero_parts_generator, 959
GiNaC::pointer_to_map_function, 966
operator(), 968
pointer_to_map_function, 968
ptr, 968
GiNaC::pointer_to_map_function_1arg< T1 >, 968
arg1, 970
operator(), 970
pointer_to_map_function_1arg, 970
ptr, 970
GiNaC::pointer_to_map_function_2args< T1, T2 >, 971
arg1, 973
arg2, 973
operator(), 972
pointer_to_map_function_2args, 972
ptr, 973
GiNaC::pointer_to_map_function_3args< T1, T2, T3 >, 973
arg1, 975
arg2, 975
arg3, 976
operator(), 975
pointer_to_map_function_3args, 975
ptr, 975
GiNaC::pointer_to_member_to_map_function< C >, 976
c, 978
operator(), 978
pointer_to_member_to_map_function, 978
ptr, 978
GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >, 979
arg1, 981
c, 981
operator(), 981
pointer_to_member_to_map_function_1arg, 981
ptr, 981
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >, 982
arg1, 984
arg2, 985
c, 984
operator(), 984
pointer_to_member_to_map_function_2args, 984
ptr, 984
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >, 985
arg1, 987
arg2, 988
arg3, 988
c, 987
operator(), 987
pointer_to_member_to_map_function_3args, 987
ptr, 987
GiNaC::pole_error, 988
deg, 990
degree, 990
pole_error, 989
GiNaC::possymbol, 990
duplicate, 996
get_domain, 996
possymbol, 995
GiNaC::power, 996
archive, 1007
basis, 1012
coeff, 1003
conjugate, 1007
degree, 1003
derivative, 1008
do_print_csrc, 1010
do_print_csrc_cl_N, 1010
do_print_dflt, 1009
do_print_latex, 1009
do_print_python, 1010
do_print_python_repr, 1010
eval, 1003
eval_ncmul, 1008
evalf, 1004
evalm, 1004
expand, 1009
expand_add, 1010
expand_add_2, 1011
expand_mul, 1011
exponent, 1012
has, 1005
imag_part, 1007
info, 1001
is_polynomial, 1002
ldegree, 1003
map, 1002
mul, 1012
nops, 1002
normal, 1006
op, 1002
power, 1001

precedence, 1001
print_power, 1009
read_archive, 1007
real_part, 1007
return_type, 1008
return_type_tinfo, 1008
series, 1005
subs, 1005
to_polynomial, 1006
to_rational, 1006
GiNaC::print_context, 1013
~print_context, 1014
options, 1015
print_context, 1014
s, 1014
GiNaC::print_context_options, 1015
get_id, 1016
get_name, 1016
get_parent_name, 1016
id, 1017
name, 1016
parent_name, 1016
print_context_options, 1016
GiNaC::print_csrc, 1017
print_csrc, 1018
GiNaC::print_csrc_cl_N, 1019
print_csrc_cl_N, 1020
GiNaC::print_csrc_double, 1021
print_csrc_double, 1022
GiNaC::print_csrc_float, 1023
print_csrc_float, 1024
GiNaC::print_dflt, 1025
print_dflt, 1026
GiNaC::print_functor, 1026
impl, 1028
is_valid, 1028
operator(), 1028
operator=, 1028
print_functor, 1027
GiNaC::print_functor_impl, 1029
~print_functor_impl, 1029
duplicate, 1029
operator(), 1029
GiNaC::print_latex, 1030
print_latex, 1031
GiNaC::print_memfun_handler< T, C >, 1032
duplicate, 1033
F, 1033
f, 1034
operator(), 1033
print_memfun_handler, 1033
GiNaC::print_options, 1034
print_index_dimensions, 1035
GiNaC::print_ptrfun_handler< T, C >, 1035
duplicate, 1037
F, 1036
f, 1037
operator(), 1037
print_ptrfun_handler, 1036
GiNaC::print_python, 1038
print_python, 1039
GiNaC::print_python_repr, 1039
print_python_repr, 1040
GiNaC::print_tree, 1041
delta_indent, 1042
print_tree, 1042
GiNaC::pseries, 1046
add_series, 1059
archive, 1056
coeff, 1053
coeffop, 1058
collect, 1053
conjugate, 1055
convert_to_poly, 1058
degree, 1053
derivative, 1057
do_print, 1061
do_print_latex, 1061
do_print_python, 1061
do_print_python_repr, 1062
do_print_tree, 1061
eval, 1054
eval_integ, 1056
evalf, 1054
evalm, 1056
expand, 1055
exponop, 1059
get_point, 1057
get_var, 1057
imag_part, 1056
is_compatible_to, 1058
is_terminating, 1058
is_zero, 1058
ldegree, 1053
mul_const, 1059
mul_series, 1059
nops, 1052
normal, 1055
op, 1052
point, 1062
power_const, 1060
precedence, 1052
print_series, 1060
pseries, 1051, 1052
read_archive, 1057
real_part, 1056
seq, 1062
series, 1054
shift_exponents, 1060
subs, 1054
var, 1062
GiNaC::psi1_SERIAL, 1063
serial, 1063
GiNaC::psi2_SERIAL, 1064
serial, 1064
GiNaC::ptr< T >, 1064

~ptr, 1066
get_pointer, 1068
makewritable, 1067
operator!=, 1067, 1068
operator<<, 1069
operator->, 1067
operator=, 1066
operator==, 1067, 1068
operator*, 1066
p, 1069
ptr, 1066
std::less< ptr< T > >, 1068
swap, 1067
GiNaC::realsymbol, 1069
conjugate, 1075
duplicate, 1075
get_domain, 1075
imag_part, 1075
real_part, 1075
realsymbol, 1074, 1075
GiNaC::refcounted, 1076
add_reference, 1078
get_refcount, 1078
refcount, 1079
refcounted, 1078
remove_reference, 1078
set_refcount, 1078
GiNaC::registered_class_options, 1079
get_id, 1080
get_name, 1080
get_parent_name, 1080
get_print_dispatch_table, 1080
name, 1081
parent_name, 1081
print_dispatch_table, 1081
print_func, 1080, 1081
registered_class_options, 1080
set_print_func, 1081
tinfo_key, 1081
GiNaC::relational, 1082
archive, 1089
calchash, 1091
canonical, 1090
do_print, 1091
do_print_python_repr, 1091
equal, 1087
eval_ncmul, 1090
greater, 1088
greater_or_equal, 1088
info, 1088
less, 1087
less_or_equal, 1087
lh, 1093
lhs, 1091
make_safe_bool, 1092
map, 1089
match_same_type, 1090
nops, 1088
not_equal, 1087
o, 1093
op, 1089
operator safe_bool, 1092
operator!, 1092
operators, 1087
precedence, 1088
read_archive, 1089
relational, 1088
return_type, 1090
return_type_tinfo, 1091
rh, 1093
rhs, 1092
safe_bool, 1087
subs, 1089
GiNaC::relational::safe_bool_helper, 1105
nonnull, 1105
GiNaC::remember_strategies, 1093
delete_cyclic, 1094
delete_lfu, 1094
delete_lru, 1094
delete_never, 1094
GiNaC::remember_table, 1094
add_entry, 1096
clear_all_entries, 1096
init_table, 1097
lookup_entry, 1096
max_assoc_size, 1097
remember_strategy, 1097
remember_table, 1096
remember_tables, 1096
show_statistics, 1096
table_size, 1097
GiNaC::remember_table_entry, 1098
access_counter, 1100
get_last_access, 1099
get_result, 1099
get_successful_hits, 1099
hashvalue, 1100
is_equal, 1099
last_access, 1100
remember_table_entry, 1099
result, 1100
seq, 1100
successful_hits, 1100
GiNaC::remember_table_list, 1101
add_entry, 1102
lookup_entry, 1102
max_assoc_size, 1102
remember_strategy, 1102
remember_table_list, 1102
GiNaC::return_type_t, 1103
operator!=, 1104
operator<, 1103
operator==, 1103
rl, 1104
tinfo, 1104
GiNaC::return_types, 1104

commutative, 1105
noncommutative, 1105
noncommutative_composite, 1105
GiNaC::scalar_products, 1105
add, 1106
add_vectors, 1106
clear, 1107
debugprint, 1107
evaluate, 1107
is_defined, 1107
spm, 1108
GiNaC::series_options, 1108
suppress_branchcut, 1108
GiNaC::solve_algo, 1109
automatic, 1109
bareiss, 1110
divfree, 1109
gauss, 1109
markowitz, 1110
GiNaC::spinidx, 1110
archive, 1117
conjugate, 1117
do_print, 1119
do_print_latex, 1119
do_print_tree, 1119
dotted, 1120
is_dotted, 1118
is_dummy_pair_same_type, 1117
is_undotted, 1118
match_same_type, 1118
read_archive, 1117
spinidx, 1116
toggle_dot, 1119
toggle_variance_dot, 1119
GiNaC::spinmetric, 1120
contract_with, 1125
do_print, 1126
do_print_latex, 1126
eval_indexed, 1125
info, 1125
GiNaC::spmapkey, 1126
debugprint, 1128
dim, 1129
operator<, 1128
operator==, 1128
spmapkey, 1128
v1, 1129
v2, 1129
GiNaC::status_flags, 1129
dynallocated, 1130
evaluated, 1130
expanded, 1130
has_indices, 1130
has_no_indices, 1130
hash_calculated, 1130
is_negative, 1130
is_positive, 1130
not_shareable, 1130
purely_indefinite, 1130
GiNaC::structure< T, ComparisonPolicy >, 1130
add_indexed, 1144
calchash, 1146
coeff, 1140
collect, 1141
contract_with, 1145
degree, 1140
derivative, 1141
eval, 1135
eval_indexed, 1136
eval_ncmul, 1136
evalm, 1136
expand, 1141
get_class_name, 1135
get_free_indices, 1144
get_struct, 1147
has, 1139
info, 1137
integer_content, 1143
is_equal_same_type, 1146
ldegree, 1140
let_op, 1138
map, 1140
match, 1139
match_same_type, 1139
max_coefficient, 1143
nops, 1137
normal, 1142
obj, 1147
op, 1137
operator->, 1146
operator[], 1138
precedence, 1137
print, 1136
return_type, 1145
return_type_tinfo, 1146
scalar_mul_indexed, 1144
series, 1142
smod, 1143
structure, 1135
subs, 1139
to_polynomial, 1143
to_rational, 1142
GiNaC::su3d, 1148
contract_with, 1152
do_print, 1153
do_print_latex, 1153
eval_indexed, 1152
return_type, 1152
GiNaC::su3f, 1153
contract_with, 1158
do_print, 1158
do_print_latex, 1158
eval_indexed, 1158
return_type, 1158
GiNaC::su3one, 1159
do_print, 1163

do_print_latex, 1163
GiNaC::su3t, 1164
contract_with, 1168
do_print, 1168
do_print_latex, 1168
GiNaC::subs_options, 1169
algebraic, 1169
no_index_renaming, 1169
no_pattern, 1169
pattern_is_not_product, 1169
pattern_is_product, 1169
really_subs_idx, 1169
subs_algebraic, 1169
subs_no_pattern, 1169
GiNaC::sy_is_less, 1170
operator(), 1170
sy_is_less, 1170
v, 1170
GiNaC::sy_swap, 1171
operator(), 1171
swapped, 1171
sy_swap, 1171
v, 1171
GiNaC::sym_desc, 1172
deg_a, 1174
deg_b, 1174
ldeg_a, 1174
ldeg_b, 1174
max_deg, 1175
max_lcnops, 1175
operator<, 1174
sym, 1174
sym_desc, 1173
GiNaC::symbol, 1175
archive, 1183
calchash, 1184
conjugate, 1182
derivative, 1184
do_print, 1186
do_print_latex, 1186
do_print_python_repr, 1186
do_print_tree, 1186
eval, 1181
evalf, 1181
get_domain, 1185
get_name, 1185
get_TeX_name, 1185
imag_part, 1183
info, 1181
is_equal_same_type, 1184
is_polynomial, 1183
name, 1187
next_serial, 1187
normal, 1182
read_archive, 1183
real_part, 1183
serial, 1187
series, 1181
set_name, 1185
set_TeX_name, 1185
subs, 1181
symbol, 1180
TeX_name, 1187
to_polynomial, 1182
to_rational, 1182
GiNaC::symbolset, 1187
has, 1188
insert_symbols, 1188
s, 1188
symbolset, 1188
GiNaC::symmetry, 1189
add, 1196
antisymmetric, 1194
archive, 1195
calchash, 1195
canonicalize, 1198
children, 1198
cyclic, 1194
do_print, 1197
do_print_tree, 1197
get_type, 1195
has_cyclic, 1197
has_nonsymmetric, 1196
has_symmetry, 1196
indices, 1198
none, 1194
read_archive, 1195
set_type, 1196
sy_is_less, 1197
sy_swap, 1197
symmetric, 1194
symmetry, 1194
symmetry_type, 1194
type, 1198
validate, 1196
GiNaC::symminfo, 1199
coeff, 1200
num, 1201
orig, 1201
symminfo, 1200
symmterm, 1200
GiNaC::symminfo_is_less_by_orig, 1201
operator(), 1201
GiNaC::symminfo_is_less_by_symmterm, 1202
operator(), 1202
GiNaC::tensdelta, 1202
contract_with, 1207
do_print, 1207
do_print_latex, 1208
eval_indexed, 1207
info, 1207
return_type, 1207
GiNaC::tensepsilon, 1208
archive, 1214
contract_with, 1213
do_print, 1214

do_print_latex, 1215
eval_indexed, 1213
info, 1213
minkowski, 1215
pos_sig, 1215
read_archive, 1214
return_type, 1214
tensepsilon, 1213
GiNaC::tensmetric, 1216
contract_with, 1221
do_print, 1221
eval_indexed, 1221
info, 1221
return_type, 1221
GiNaC::tensor, 1222
replace_contr_index, 1226
return_type, 1226
GiNaC::terminfo, 1227
orig, 1228
symm, 1228
terminfo, 1228
GiNaC::terminfo_is_less, 1229
operator(), 1229
GiNaC::unarchive_table_t, 1234
∼unarchive_table_t, 1234
find, 1234
insert, 1234
unarch_map, 1235
unarchive_table_t, 1234
usecount, 1235
GiNaC::user_defined_kernel, 1235
do_print, 1242
f, 1242
is_numeric, 1241
Laurent_series, 1241
let_op, 1241
nops, 1241
op, 1241
user_defined_kernel, 1241
uses_Laurent_series, 1242
x, 1242
GiNaC::varidx, 1243
archive, 1249
covariant, 1251
do_print, 1251
do_print_tree, 1251
is_contravariant, 1250
is_covariant, 1250
is_dummy_pair_same_type, 1249
match_same_type, 1250
read_archive, 1249
toggle_variance, 1251
varidx, 1249
GiNaC::visitor, 1252
∼visitor, 1252
GiNaC::wildcard, 1252
archive, 1257
calchash, 1257
do_print, 1258
do_print_python_repr, 1258
do_print_tree, 1258
get_label, 1257
label, 1258
match, 1257
read_archive, 1257
wildcard, 1256
GiNaC::zeta1_SERIAL, 1259
serial, 1259
GiNaC::zeta2_SERIAL, 1259
serial, 1260
GINAC_ASSERT
assertion.h, 1266
GINAC_BIND_UNARCHIVER
archive.h, 1265
GiNaC, 93, 99, 113, 119, 134–136, 139, 204, 207–210, 212, 218, 219, 235, 269, 271–274, 282, 292, 295
GINAC_DECLARE_PRINT_CONTEXT
print.h, 1356
GINAC_DECLARE_PRINT_CONTEXT_BASE
print.h, 1356
GINAC_DECLARE_PRINT_CONTEXT_COMMON
print.h, 1356
GINAC_DECLARE_REGISTERED_CLASS
registrar.h, 1361
GINAC_DECLARE_REGISTERED_CLASS_COMMON
registrar.h, 1360
GINAC_DECLARE_REGISTERED_CLASS_NO_CTORS
registrar.h, 1360
GINAC_DECLARE_UNARCHIVER
archive.h, 1264
GiNaC, 93, 111, 112, 118, 119, 134, 135, 137, 143, 204, 210–212, 214, 218, 219, 253, 269, 271, 273, 274, 278, 286, 287, 292
GINAC_IMPLEMENT_PRINT_CONTEXT
print.h, 1356
GINAC_IMPLEMENT_REGISTERED_CLASS
registrar.h, 1361
GINAC_IMPLEMENT_REGISTERED_CLASS_OPT
GiNaC, 93, 95, 98, 112, 118, 132–135, 139, 203, 207–210, 212, 217, 218, 234, 269, 270, 272–274, 281, 292
registrar.h, 1361
GINAC_IMPLEMENT_REGISTERED_CLASS_OPT_T
registrar.h, 1362
GINAC_LT_AGE
version.h, 1379
GINAC_LT_CURRENT
version.h, 1378
GINAC_LT_REVISION
version.h, 1379
GINACLIB_ARCHIVE_AGE
version.h, 1379
GINACLIB_ARCHIVE_VERSION
version.h, 1379
GINACLIB_MAJOR_VERSION

version.h, 1378
GINACLIB_MICRO_VERSION
version.h, 1378
GINACLIB_MINOR_VERSION
version.h, 1378
GINACLIB_STR
version.h, 1379
GINACLIB_STR_HELPER
version.h, 1379
GINACLIB_VERSION
version.h, 1379
golden_ratio_hash
GiNaC, 288
greater
GiNaC::relational, 1088
greater_or_equal
GiNaC::relational, 1088
Gt
GiNaC::Gt, 713
GiNaC::Gt::kernel, 786
Gt.cpp, 1307
DEBUG_GT, 1308
Gt.h, 1308
Gt_helpers.cpp, 1308
Gt_helpers.h, 1309
guess_precision
GiNaC, 244

H_deriv
GiNaC, 180
H_eval
GiNaC, 180
H_evalf
GiNaC, 179
H_print_latex
GiNaC, 180
H_series
GiNaC, 180
handle_factor
GiNaC::make_flat_inserter, 810
has
GiNaC, 121
GiNaC::basic, 369
GiNaC::ex, 574
GiNaC::Gt, 716
GiNaC::mul, 862
GiNaC::numeric, 935
GiNaC::power, 1005
GiNaC::structure< T, ComparisonPolicy >, 1139
GiNaC::symbolset, 1188
has_cyclic
GiNaC::symmetry, 1197
has_derivative
GiNaC::function_options, 699
has_dummy_index_for
GiNaC::indexed, 756
has_ex
GiNaC::archive_node, 353
has_expression
GiNaC::archive_node, 354
has_indices
GiNaC::info_flags, 761
GiNaC::status_flags, 1130
has_no_indices
GiNaC::status_flags, 1130
has_nonsymmetric
GiNaC::symmetry, 1196
has_power
GiNaC::function_options, 700
has_same_ex_as
GiNaC::archive_node, 352
has_symmetry
GiNaC::symmetry, 1196
has_trailing_zero
GiNaC::integration_kernel, 778
hash_calculated
GiNaC::status_flags, 1130
hash_map.h, 1310
hash_seed.h, 1310
hashvalue
GiNaC::basic, 387
GiNaC::remember_table_entry, 1100
hasindex
GiNaC, 141
haswild
GiNaC, 292
head
GiNaC::composition_generator::coolmulti, 486
heur_gcd
GiNaC, 227
heur_gcd_z
GiNaC, 226
hold
GiNaC::basic, 383
hold_ncmul
GiNaC, 219
GiNaC::ncmul, 923

I
GiNaC, 295
i
GiNaC::composition_generator::coolmulti, 486
GiNaC::const_iterator, 448
GiNaC::internal::_iter_rep, 316
i_end
GiNaC::internal::_iter_rep, 317
id
GiNaC::print_context_options, 1017
identify_parents
GiNaC::class_info< OPT >, 405
idx
GiNaC, 295
GiNaC::idx, 733
GiNaC::info_flags, 761
idx.cpp, 1311
idx.h, 1312
idx_symmetrization
GiNaC, 141

ifactor
GiNaC, 205
imag
GiNaC, 257
GiNaC::numeric, 951
imag_part
GiNaC, 121
GiNaC::add, 332
GiNaC::basic, 378
GiNaC::constant, 462
GiNaC::container< C >, 476
GiNaC::ex, 574
GiNaC::function, 659
GiNaC::indexed, 753
GiNaC::matrix, 823
GiNaC::mul, 863
GiNaC::ncmul, 921
GiNaC::numeric, 939
GiNaC::power, 1007
GiNaC::pseries, 1056
GiNaC::realsymbol, 1075
GiNaC::symbol, 1183
imag_part_conjugate
GiNaC, 148
imag_part_eval
GiNaC, 148
imag_part_evalf
GiNaC, 147
imag_part_expl_derivative
GiNaC, 149
imag_part_f
GiNaC::function_options, 702
imag_part_func
GiNaC::function_options, 680–682, 695
imag_part_funcp
GiNaC, 70
imag_part_funcp_1
GiNaC, 71
imag_part_funcp_10
GiNaC, 83
imag_part_funcp_11
GiNaC, 84
imag_part_funcp_12
GiNaC, 86
imag_part_funcp_13
GiNaC, 87
imag_part_funcp_14
GiNaC, 89
imag_part_funcp_2
GiNaC, 73
imag_part_funcp_3
GiNaC, 74
imag_part_funcp_4
GiNaC, 75
imag_part_funcp_5
GiNaC, 76
imag_part_funcp_6
GiNaC, 78
imag_part_funcp_7
GiNaC, 79
imag_part_funcp_8
GiNaC, 80
imag_part_funcp_9
GiNaC, 82
imag_part_funcp_exvector
GiNaC, 90
imag_part_imag_part
GiNaC, 148
imag_part_print_latex
GiNaC, 148
imag_part_real_part
GiNaC, 148
imag_part_use_exvector_args
GiNaC::function_options, 704
impl
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1231
GiNaC::print_functor, 1028
increment
GiNaC::const_postorder_iterator, 451
GiNaC::const_preorder_iterator, 454
indefinite
GiNaC::info_flags, 761
index0
GiNaC, 275
index1
GiNaC, 275
index2
GiNaC, 275
index3
GiNaC, 275
index_dimensions
GiNaC, 268
indexed
GiNaC::indexed, 748–752
GiNaC::info_flags, 761
indexed.cpp, 1313
indexed.h, 1315
indices
GiNaC::symmetry, 1198
indices_consistent
GiNaC, 139
info
GiNaC::add, 327
GiNaC::basic, 367
GiNaC::class_info< OPT >::tree_node, 1233
GiNaC::constant, 461
GiNaC::container< C >, 473
GiNaC::ex, 571
GiNaC::expairseq, 610
GiNaC::function, 660
GiNaC::idx, 734
GiNaC::indexed, 752
GiNaC::minkmetric, 841
GiNaC::mul, 860
GiNaC::ncmul, 918

GiNaC::numeric, 934
GiNaC::power, 1001
GiNaC::relational, 1088
GiNaC::spinmetric, 1125
GiNaC::structure< T, ComparisonPolicy >, 1137
GiNaC::symbol, 1181
GiNaC::tensdelta, 1207
GiNaC::tensepsilon, 1213
GiNaC::tensmetric, 1221
info_f
GiNaC::function_options, 702
info_func
GiNaC::function_options, 692–694, 696
info_funcp
GiNaC, 71
info_funcp_1
GiNaC, 72
info_funcp_10
GiNaC, 84
info_funcp_11
GiNaC, 85
info_funcp_12
GiNaC, 87
info_funcp_13
GiNaC, 88
info_funcp_14
GiNaC, 90
info_funcp_2
GiNaC, 73
info_funcp_3
GiNaC, 74
info_funcp_4
GiNaC, 76
info_funcp_5
GiNaC, 77
info_funcp_6
GiNaC, 78
info_funcp_7
GiNaC, 80
info_funcp_8
GiNaC, 81
info_funcp_9
GiNaC, 82
info_funcp_exvector
GiNaC, 91
info_use_exvector_args
GiNaC::function_options, 705
inifcns.cpp, 1316
inifcns.h, 1319
inifcns_elliptic.cpp, 1320
inifcns_gamma.cpp, 1322
inifcns_nstdsums.cpp, 1323
inifcns_trans.cpp, 1325
init
GiNaC::basic_multi_iterator< T >, 398
GiNaC::multi_iterator_counter< T >, 874
GiNaC::multi_iterator_counter_indv< T >, 878
GiNaC::multi_iterator_ordered< T >, 882
GiNaC::multi_iterator_ordered_eq< T >, 886
GiNaC::multi_iterator_ordered_eq_indv< T >, 889
GiNaC::multi_iterator_permutation< T >, 894
GiNaC::multi_iterator_shuffle< T >, 898
GiNaC::multi_iterator_shuffle_prime< T >, 902
init_table
GiNaC::remember_table, 1097
init_unarchivers
GiNaC::library_init, 808
initialize
GiNaC::function_options, 672
insert
GiNaC::unarchive_table_t, 1234
insert_symbols
GiNaC::symbolset, 1188
int_length
GiNaC::numeric, 952
integer
GiNaC::info_flags, 760
integer_content
GiNaC::add, 330
GiNaC::basic, 374
GiNaC::ex, 583
GiNaC::mul, 864
GiNaC::numeric, 938
GiNaC::structure< T, ComparisonPolicy >, 1143
integer_partition
GiNaC::Gt_detail, 312
integer_polynomial
GiNaC::info_flags, 761
integral
GiNaC::error_and_integral, 557
GiNaC::integral, 767
integral.cpp, 1328
integral.h, 1329
integrate
GiNaC::Gt_detail::pathintegral_term, 962
integration_kernel.cpp, 1329
integration_kernel.h, 1331
interpolate
GiNaC, 226
inv_at_cit
GiNaC::archive, 338
inverse
GiNaC, 216, 253
GiNaC::matrix, 829, 830
GiNaC::numeric, 944
inverse_atoms
GiNaC::archive, 344
iquo
GiNaC, 250
irem
GiNaC, 249
is_a
GiNaC, 95, 128, 270
GiNaC::ex, 594
is_canonical
GiNaC::expairseq, 620

is_canonical_numeric
GiNaC::expair, 600
is_cinteger
GiNaC, 256
GiNaC::numeric, 948
is_clifford_tinfo
GiNaC, 112
is_color_tinfo
GiNaC, 116
is_compatible_to
GiNaC::pseries, 1058
is_contravariant
GiNaC::varidx, 1250
is_covariant
GiNaC::varidx, 1250
is_crational
GiNaC, 256
GiNaC::numeric, 948
is_defined
GiNaC::scalar_products, 1107
is_dim_numeric
GiNaC::idx, 738
is_dim_symbolic
GiNaC::idx, 738
is_dirac_slash
GiNaC, 100
is_discriminant_of_quadratic_number_field
GiNaC, 205
is_dotted
GiNaC::spinidx, 1118
is_dummy_pair
GiNaC, 136
is_dummy_pair_same_type
GiNaC::idx, 737
GiNaC::spinidx, 1117
GiNaC::varidx, 1249
is_equal
GiNaC::basic, 382
GiNaC::ex, 587
GiNaC::expair, 600
GiNaC::numeric, 945
GiNaC::remember_table_entry, 1099
is_equal_same_type
GiNaC::basic, 379
GiNaC::constant, 463
GiNaC::container< C >, 476
GiNaC::expairseq, 613
GiNaC::fderivative, 641
GiNaC::function, 660
GiNaC::numeric, 940
GiNaC::structure< T, ComparisonPolicy >, 1146
GiNaC::symbol, 1184
is_even
GiNaC, 255
GiNaC::numeric, 947
is_ex_the_function
function.h, 1305
is_exactly_a
GiNaC, 96, 128
GiNaC::ex, 594
is_integer
GiNaC, 254
GiNaC::numeric, 946
is_less
GiNaC::expair, 600
is_negative
GiNaC, 254
GiNaC::numeric, 946
GiNaC::status_flags, 1130
is_nonneg_integer
GiNaC, 255
GiNaC::numeric, 947
is_numeric
GiNaC::Ebar_kernel, 523
GiNaC::Eisenstein_h_kernel, 533
GiNaC::Eisenstein_kernel, 543
GiNaC::ELi_kernel, 553
GiNaC::idx, 737
GiNaC::integration_kernel, 778
GiNaC::Kronecker_dtau_kernel, 793
GiNaC::Kronecker_dz_kernel, 802
GiNaC::modular_form_kernel, 850
GiNaC::multiple_polylog_kernel, 909
GiNaC::user_defined_kernel, 1241
is_odd
GiNaC, 255
GiNaC::numeric, 947
is_order_function
GiNaC, 166
is_polynomial
GiNaC, 121
GiNaC::add, 328
GiNaC::basic, 371
GiNaC::constant, 461
GiNaC::ex, 577
GiNaC::mul, 861
GiNaC::numeric, 934
GiNaC::power, 1002
GiNaC::symbol, 1183
is_pos_integer
GiNaC, 255
GiNaC::numeric, 947
is_positive
GiNaC, 254
GiNaC::numeric, 946
GiNaC::status_flags, 1130
is_prime
GiNaC, 255
GiNaC::numeric, 947
is_rational
GiNaC, 256
GiNaC::numeric, 948
is_real
GiNaC, 256
GiNaC::numeric, 948
is_symbolic

GiNaC::idx, 737
is_terminating
GiNaC, 271
GiNaC::pseries, 1058
is_the_function
GiNaC, 135
is_the_function< G_SERIAL >
GiNaC, 164
is_the_function< iterated_integral_SERIAL >
GiNaC, 166
is_the_function< psi_SERIAL >
GiNaC, 165
is_the_function< zeta_SERIAL >
GiNaC, 164
is_undotted
GiNaC::spinidx, 1118
is_valid
GiNaC::print_functor, 1028
is_zero
GiNaC, 127, 254
GiNaC::ex, 588
GiNaC::numeric, 946
GiNaC::pseries, 1058
is_zero_matrix
GiNaC::ex, 588
GiNaC::matrix, 831
isqrt
GiNaC, 252
iterated_integral
GiNaC, 165
iterated_integral2_eval
GiNaC, 169
iterated_integral2_evalf
GiNaC, 169
iterated_integral3_eval
GiNaC, 169
iterated_integral3_evalf
GiNaC, 169
iterated_integral_evalf_impl
GiNaC, 169
iterator_category
GiNaC::const_iterator, 444
GiNaC::const_postorder_iterator, 449
GiNaC::const_preorder_iterator, 452

K
GiNaC::Eisenstein_kernel, 545
GiNaC::Kronecker_dtau_kernel, 795
GiNaC::Kronecker_dz_kernel, 804
k
GiNaC::Eisenstein_h_kernel, 535
GiNaC::Eisenstein_kernel, 544
GiNaC::modular_form_kernel, 851
kernel
GiNaC::Gt::kernel, 786
Kronecker_dtau_kernel
GiNaC::Kronecker_dtau_kernel, 793
Kronecker_dz_kernel
GiNaC::Kronecker_dz_kernel, 802
kronecker_symbol
GiNaC, 205

label
GiNaC::wildcard, 1258
lanczos_coeffs
GiNaC::lanczos_coeffs, 805
laplace
GiNaC::determinant_algo, 489
last_access
GiNaC::remember_table_entry, 1100
latex
GiNaC, 267
latex_name
GiNaC::function_options, 672
Laurent_series
GiNaC::Eisenstein_h_kernel, 533
GiNaC::Eisenstein_kernel, 543
GiNaC::integration_kernel, 779
GiNaC::modular_form_kernel, 850
GiNaC::user_defined_kernel, 1241
lcm
GiNaC, 230, 251
lcm_of_coefficients_denominators
GiNaC, 221
lcmcoeff
GiNaC, 221
lcoeff
GiNaC::ex, 578
ldeg_a
GiNaC::sym_desc, 1174
ldeg_b
GiNaC::sym_desc, 1174
ldegree
GiNaC, 122
GiNaC::add, 328
GiNaC::basic, 371
GiNaC::ex, 577
GiNaC::integral, 768
GiNaC::mul, 861
GiNaC::ncmul, 918
GiNaC::numeric, 935
GiNaC::power, 1003
GiNaC::pseries, 1053
GiNaC::structure< T, ComparisonPolicy >, 1140
less
GiNaC::relational, 1087
less_or_equal
GiNaC::relational, 1087
let_op
GiNaC::basic, 368
GiNaC::clifford, 418
GiNaC::container< C >, 474
GiNaC::Ebar_kernel, 523
GiNaC::Eisenstein_h_kernel, 532
GiNaC::Eisenstein_kernel, 543
GiNaC::ELi_kernel, 553
GiNaC::ex, 573
GiNaC::integral, 769

GiNaC::Kronecker_dtau_kernel, 793
GiNaC::Kronecker_dz_kernel, 802
GiNaC::matrix, 821
GiNaC::modular_form_kernel, 849
GiNaC::multiple_polylog_kernel, 909
GiNaC::structure< T, ComparisonPolicy >, 1138
GiNaC::user_defined_kernel, 1241
lgamma
GiNaC, 244
lgamma_conjugate
GiNaC, 171
lgamma_deriv
GiNaC, 170
lgamma_eval
GiNaC, 170
lgamma_evalf
GiNaC, 170
lgamma_series
GiNaC, 170
lh
GiNaC::relational, 1093
lhs
GiNaC, 126
GiNaC::ex, 573
GiNaC::relational, 1091
Li2
GiNaC, 243
Li2_
GiNaC, 243
Li2_conjugate
GiNaC, 157
Li2_deriv
GiNaC, 156
Li2_eval
GiNaC, 156
Li2_evalf
GiNaC, 156
Li2_projection
GiNaC, 243
Li2_series
GiNaC, 156, 243
Li3_eval
GiNaC, 157
Li_deriv
GiNaC, 177
Li_eval
GiNaC, 177
Li_evalf
GiNaC, 176
Li_print_latex
GiNaC, 177
Li_series
GiNaC, 177
library_init
GiNaC::library_init, 808
library_initializer
GiNaC, 294
likely
compiler.h, 1278
link_ex
GiNaC, 131, 132
list
GiNaC::info_flags, 761
log
GiNaC, 237
log2
GiNaC, 287
log_conjugate
GiNaC, 186
log_deriv
GiNaC, 185
log_eval
GiNaC, 185
log_evalf
GiNaC, 184
log_expand
GiNaC, 186
log_imag_part
GiNaC, 185
log_info
GiNaC, 186
log_real_part
GiNaC, 185
log_series
GiNaC, 185
lookup_entry
GiNaC::remember_table, 1096
GiNaC::remember_table_list, 1102
lookup_map
GiNaC, 91
lookup_remember_table
GiNaC::function, 662
lorentz_eps
GiNaC, 285
lorentz_g
GiNaC, 283
lsolve
GiNaC, 162
lst
GiNaC, 91
lst.cpp, 1332
lst.h, 1333
lst_evaluate
GiNaC::Gt, 720, 721
lst_to_clifford
GiNaC, 107, 108
lst_to_matrix
GiNaC, 213

m
GiNaC::basic_partition_generator::mpartition2, 853
GiNaC::Ebar_kernel, 525
GiNaC::ELi_kernel, 554
GiNaC::matrix, 835
GiNaC::partition_with_zero_parts_generator, 959
make_flat

GiNaC::expairseq, 619
make_flat_inserter
GiNaC::make_flat_inserter, 810
make_real_float
GiNaC, 234
make_return_type_t
GiNaC, 272
make_safe_bool
GiNaC::relational, 1092
makewritable
GiNaC::ptr< T >, 1067
makewriteable
GiNaC::ex, 592
map
GiNaC::basic, 370
GiNaC::ex, 576
GiNaC::expairseq, 611
GiNaC::idx, 734
GiNaC::power, 1002
GiNaC::relational, 1089
GiNaC::structure< T, ComparisonPolicy >, 1140
map_eval_integ
GiNaC, 293
map_evalm
GiNaC, 293
markowitz
GiNaC::solve_algo, 1110
markowitz_elimination
GiNaC::matrix, 833
match
GiNaC, 125
GiNaC::basic, 369
GiNaC::ex, 575
GiNaC::expairseq, 612
GiNaC::Gt, 716
GiNaC::structure< T, ComparisonPolicy >, 1139
GiNaC::wildcard, 1257
match_same_type
GiNaC::basic, 369
GiNaC::clifford, 415
GiNaC::color, 435
GiNaC::fderivative, 642
GiNaC::function, 661
GiNaC::idx, 736
GiNaC::matrix, 824
GiNaC::relational, 1090
GiNaC::spinidx, 1118
GiNaC::structure< T, ComparisonPolicy >, 1139
GiNaC::varidx, 1250
matrix
GiNaC::matrix, 819, 820
matrix.cpp, 1333
matrix.h, 1334
max_assoc_size
GiNaC::remember_table, 1097
GiNaC::remember_table_list, 1102
max_coefficient
GiNaC::add, 331
GiNaC::basic, 375
GiNaC::ex, 586
GiNaC::mul, 864
GiNaC::numeric, 938
GiNaC::structure< T, ComparisonPolicy >, 1143
max_deg
GiNaC::sym_desc, 1175
max_integration_level
GiNaC::integral, 772
max_lcnops
GiNaC::sym_desc, 1175
metric
GiNaC::clifford, 419
metric_tensor
GiNaC, 283
minimal_dim
GiNaC, 137
GiNaC::idx, 738
minkmetric
GiNaC::minkmetric, 841
minkowski
GiNaC::tensepsilon, 1215
mod
GiNaC, 248
modular_form_kernel
GiNaC::modular_form_kernel, 849
mpartition2
GiNaC::basic_partition_generator::mpartition2, 852
mpgen
GiNaC::basic_partition_generator, 402
mul
GiNaC::add, 336
GiNaC::matrix, 825, 826
GiNaC::mul, 859, 860
GiNaC::numeric, 941
GiNaC::power, 1012
mul.cpp, 1336
mul.h, 1336
mul_const
GiNaC::pseries, 1059
mul_dyn
GiNaC::numeric, 942
mul_scalar
GiNaC::matrix, 826
mul_series
GiNaC::pseries, 1059
multi_iterator_counter
GiNaC::multi_iterator_counter< T >, 874
multi_iterator_counter_indv
GiNaC::multi_iterator_counter_indv< T >, 877, 878
multi_iterator_ordered
GiNaC::multi_iterator_ordered< T >, 881, 882
multi_iterator_ordered_eq
GiNaC::multi_iterator_ordered_eq< T >, 885
multi_iterator_ordered_eq_indv
GiNaC::multi_iterator_ordered_eq_indv< T >, 889

multi_iterator_permutation
GiNaC::multi_iterator_permutation< T >, 893
multi_iterator_shuffle
GiNaC::multi_iterator_shuffle< T >, 897
multi_iterator_shuffle_prime
GiNaC::multi_iterator_shuffle_prime< T >, 902
multinomial_coefficient
GiNaC, 287
multiple_polylog_kernel
GiNaC::multiple_polylog_kernel, 909
multiply_lcm
GiNaC, 221
my_ios_callback
GiNaC, 265
my_ios_index
GiNaC, 265

N
GiNaC::basic_multi_iterator< T >, 399
GiNaC::Eisenstein_h_kernel, 535
GiNaC::Eisenstein_kernel, 544
n
GiNaC::basic_partition_generator::mpartition2, 853
GiNaC::Ebar_kernel, 525
GiNaC::ELi_kernel, 554
GiNaC::Kronecker_dtau_kernel, 795
GiNaC::Kronecker_dz_kernel, 804
N_internal
GiNaC::multi_iterator_shuffle< T >, 898
name
GiNaC::archive::archived_ex, 358
GiNaC::archive_node::property, 1044
GiNaC::archive_node::property_info, 1045
GiNaC::constant, 465
GiNaC::function_options, 701
GiNaC::print_context_options, 1016
GiNaC::registered_class_options, 1081
GiNaC::symbol, 1187
nargs
GiNaC::Gt, 723
ncmul
GiNaC::mul, 871
GiNaC::ncmul, 917
ncmul.cpp, 1337
ncmul.h, 1338
negative
GiNaC::info_flags, 760
negint
GiNaC::info_flags, 760
next
GiNaC::class_info< OPT >, 405
GiNaC::composition_generator, 441
GiNaC::composition_generator::coolmulti::element, 546
GiNaC::partition_generator, 956
GiNaC::partition_with_zero_parts_generator, 959
next_partition
GiNaC::basic_partition_generator::mpartition2, 852
next_permutation
GiNaC::composition_generator::coolmulti, 486
next_print_context_id
GiNaC, 296
next_serial
GiNaC::constant, 465
GiNaC::symbol, 1187
ni
GiNaC::Gt::kernel, 787
no_heur_gcd
GiNaC::gcd_options, 707
no_index_dimensions
GiNaC, 268
no_index_renaming
GiNaC::subs_options, 1169
no_part_factored
GiNaC::gcd_options, 708
no_pattern
GiNaC::subs_options, 1169
no_type
GiNaC::has_distance< T >, 725
nodes
GiNaC::archive, 343
noncommutative
GiNaC::return_types, 1105
noncommutative_composite
GiNaC::return_types, 1105
none
GiNaC::symmetry, 1194
nonnegative
GiNaC::info_flags, 760
nonnegint
GiNaC::info_flags, 760
nonnull
GiNaC::relational::safe_bool_helper, 1105
nops
GiNaC, 120, 214
GiNaC::basic, 367
GiNaC::clifford, 417
GiNaC::container< C >, 473
GiNaC::Ebar_kernel, 523
GiNaC::Eisenstein_h_kernel, 532
GiNaC::Eisenstein_kernel, 542
GiNaC::ELi_kernel, 553
GiNaC::ex, 571
GiNaC::expairseq, 610
GiNaC::Gt, 715
GiNaC::idx, 734
GiNaC::integral, 768
GiNaC::Kronecker_dtau_kernel, 793
GiNaC::Kronecker_dz_kernel, 802
GiNaC::matrix, 821
GiNaC::modular_form_kernel, 849
GiNaC::multiple_polylog_kernel, 909
GiNaC::power, 1002
GiNaC::pseries, 1052

GiNaC::relational, 1088
GiNaC::structure< T, ComparisonPolicy >, 1137
GiNaC::user_defined_kernel, 1241
normal
GiNaC, 123
GiNaC::add, 330
GiNaC::basic, 373
GiNaC::ex, 580
GiNaC::mul, 863
GiNaC::numeric, 937
GiNaC::power, 1006
GiNaC::pseries, 1055
GiNaC::structure< T, ComparisonPolicy >, 1142
GiNaC::symbol, 1182
normal.cpp, 1339
FAST_COMPARE, 1341
STATISTICS, 1341
USE_REMEMBER, 1341
USE_TRIAL_DIVISION, 1341
normal.h, 1341
not_equal
GiNaC::relational, 1087
not_shareable
GiNaC::status_flags, 1130
not_symmetric
GiNaC, 275
nparams
GiNaC::function_options, 701
nqexand_transformer
GiNaC::Gt, 724
num
GiNaC::symminfo, 1201
num_expressions
GiNaC::archive, 340
number
GiNaC::constant, 465
number_of_type
GiNaC, 139
numer
GiNaC, 122, 257
GiNaC::ex, 581
GiNaC::numeric, 951
numer_denom
GiNaC, 122
GiNaC::ex, 582
numeric
GiNaC::info_flags, 760
GiNaC::numeric, 932, 933
numeric.cpp, 1343
numeric.h, 1346
Nv
GiNaC::multi_iterator_counter_indv< T >, 879
GiNaC::multi_iterator_ordered_eq_indv< T >, 890

o
GiNaC::relational, 1093
obj
GiNaC::structure< T, ComparisonPolicy >, 1147
odd
GiNaC::info_flags, 760
op
GiNaC, 126
GiNaC::basic, 367
GiNaC::clifford, 418
GiNaC::container< C >, 474
GiNaC::Ebar_kernel, 523
GiNaC::Eisenstein_h_kernel, 532
GiNaC::Eisenstein_kernel, 543
GiNaC::ELi_kernel, 553
GiNaC::ex, 572
GiNaC::expairseq, 611
GiNaC::idx, 734
GiNaC::integral, 768
GiNaC::Kronecker_dtau_kernel, 793
GiNaC::Kronecker_dz_kernel, 802
GiNaC::matrix, 821
GiNaC::modular_form_kernel, 849
GiNaC::multiple_polylog_kernel, 909
GiNaC::power, 1002
GiNaC::pseries, 1052
GiNaC::relational, 1089
GiNaC::structure< T, ComparisonPolicy >, 1137
GiNaC::user_defined_kernel, 1241
operator long
GiNaC::_numeric_digits, 318
operator safe_bool
GiNaC::relational, 1092
operator!
GiNaC::relational, 1092
operator!=
GiNaC, 264
GiNaC::const_iterator, 447
GiNaC::const_postorder_iterator, 451
GiNaC::const_preorder_iterator, 454
GiNaC::internal::_iter_rep, 316
GiNaC::numeric, 949
GiNaC::ptr< T >, 1067, 1068
GiNaC::return_type_t, 1104
operator<
GiNaC, 264
GiNaC::const_iterator, 447
GiNaC::Gt_detail, 313
GiNaC::Gt_detail::pathintegral_term, 963
GiNaC::numeric, 949
GiNaC::return_type_t, 1103
GiNaC::spmapkey, 1128
GiNaC::sym_desc, 1174
operator<<
GiNaC, 94, 119, 120, 253, 266, 289–291
GiNaC::archive, 343
GiNaC::archive_node, 354
GiNaC::basic_multi_iterator< T >, 399
GiNaC::Gt_detail, 313
GiNaC::Gt_detail::pathintegral_term, 962
GiNaC::multi_iterator_counter< T >, 875
GiNaC::multi_iterator_counter_indv< T >, 879
GiNaC::multi_iterator_ordered< T >, 883

GiNaC::multi_iterator_ordered_eq< T >, 886
GiNaC::multi_iterator_ordered_eq_indv< T >, 890
GiNaC::multi_iterator_permutation< T >, 895
GiNaC::multi_iterator_shuffle< T >, 898
GiNaC::multi_iterator_shuffle_prime< T >, 902
GiNaC::ptr< T >, 1069
operator<=
GiNaC, 264
GiNaC::const_iterator, 447
GiNaC::numeric, 949
operator>
GiNaC, 265
GiNaC::const_iterator, 447
GiNaC::numeric, 949
operator>>
GiNaC, 94, 95, 266
GiNaC::archive, 343
GiNaC::archive_node, 354
operator>=
GiNaC, 265
GiNaC::const_iterator, 447
GiNaC::numeric, 950
operator()
GiNaC::basic_multi_iterator< T >, 398
GiNaC::derivative_map_function, 488
GiNaC::error_and_integral_is_less, 557
GiNaC::eval_integ_map_function, 559
GiNaC::evalf_map_function, 560
GiNaC::evalm_map_function, 562
GiNaC::ex_base_is_less, 595
GiNaC::ex_is_equal, 596
GiNaC::ex_is_less, 596
GiNaC::ex_swap, 597
GiNaC::expair_is_less, 602
GiNaC::expair_rest_is_less, 603
GiNaC::expair_swap, 603
GiNaC::expand_map_function, 624
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1230
GiNaC::idx_is_equal_ignore_dim, 741
GiNaC::is_not_a_clifford, 782
GiNaC::is_summation_idx, 783
GiNaC::map_function, 813
GiNaC::matrix, 826, 827
GiNaC::normal_map_function, 925
GiNaC::op0_is_equal, 954
GiNaC::pointer_to_map_function, 968
GiNaC::pointer_to_map_function_1arg< T1 >, 970
GiNaC::pointer_to_map_function_2args< T1, T2 >, 972
GiNaC::pointer_to_map_function_3args< T1, T2, T3 >, 975
GiNaC::pointer_to_member_to_map_function< C >, 978
GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >, 981
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >, 984
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >, 987
GiNaC::print_functor, 1028
GiNaC::print_functor_impl, 1029
GiNaC::print_memfun_handler< T, C >, 1033
GiNaC::print_ptrfun_handler< T, C >, 1037
GiNaC::sy_is_less, 1170
GiNaC::sy_swap, 1171
GiNaC::symminfo_is_less_by_orig, 1201
GiNaC::symminfo_is_less_by_symmterm, 1202
GiNaC::terminfo_is_less, 1229
std::equal_to< GiNaC::ex >, 555
std::hash< GiNaC::ex >, 727
std::less< GiNaC::ptr< T > >, 806
operator+
GiNaC, 258, 259, 261, 262
GiNaC::const_iterator, 446, 448
operator++
GiNaC, 262, 263
GiNaC::basic_multi_iterator< T >, 398
GiNaC::const_iterator, 445
GiNaC::const_postorder_iterator, 450
GiNaC::const_preorder_iterator, 453
GiNaC::multi_iterator_counter< T >, 874
GiNaC::multi_iterator_counter_indv< T >, 878
GiNaC::multi_iterator_ordered< T >, 882
GiNaC::multi_iterator_ordered_eq< T >, 886
GiNaC::multi_iterator_ordered_eq_indv< T >, 889
GiNaC::multi_iterator_permutation< T >, 894
GiNaC::multi_iterator_shuffle< T >, 898
operator+=
GiNaC, 260, 261
GiNaC::const_iterator, 446
operator-
GiNaC, 259, 261, 262
GiNaC::const_iterator, 446, 448
operator->
GiNaC::const_iterator, 445
GiNaC::const_postorder_iterator, 450
GiNaC::const_preorder_iterator, 453
GiNaC::ptr< T >, 1067
GiNaC::structure< T, ComparisonPolicy >, 1146
operator--
GiNaC, 262–264
GiNaC::const_iterator, 446
operator-=
GiNaC, 260, 261
GiNaC::const_iterator, 446
operator/
GiNaC, 259, 260
operator/=
GiNaC, 260, 261
operator=
GiNaC::_numeric_digits, 318
GiNaC::archive_node, 348
GiNaC::basic, 364

GiNaC::numeric, 943, 944
GiNaC::print_functor, 1028
GiNaC::ptr< T >, 1066
operator==
GiNaC, 264
GiNaC::const_iterator, 446
GiNaC::const_postorder_iterator, 451
GiNaC::const_preorder_iterator, 454
GiNaC::internal::_iter_rep, 316
GiNaC::numeric, 949
GiNaC::ptr< T >, 1067, 1068
GiNaC::return_type_t, 1103
GiNaC::spmapkey, 1128
operator[]
GiNaC::basic, 368, 369
GiNaC::basic_multi_iterator< T >, 397, 398
GiNaC::const_iterator, 445
GiNaC::ex, 572, 573
GiNaC::structure< T, ComparisonPolicy >, 1138
operator*
GiNaC, 259, 260
GiNaC::const_iterator, 445
GiNaC::const_postorder_iterator, 450
GiNaC::const_preorder_iterator, 453
GiNaC::ptr< T >, 1066
operator*=
GiNaC, 260, 261
operators
GiNaC::relational, 1087
operators.cpp, 1349
operators.h, 1351
options
GiNaC::class_info< OPT >, 405
GiNaC::expand_map_function, 624
GiNaC::print_context, 1015
Order_conjugate
GiNaC, 161
Order_eval
GiNaC, 161
Order_expl_derivative
GiNaC, 162
Order_imag_part
GiNaC, 162
Order_power
GiNaC, 162
Order_real_part
GiNaC, 161
Order_series
GiNaC, 161
orig
GiNaC::symminfo, 1201
GiNaC::terminfo, 1228
overall_coeff
GiNaC::expairseq, 621
overflow
GiNaC::basic_multi_iterator< T >, 397
overloaded
GiNaC::function_options, 699
P
GiNaC::modular_form_kernel, 851
p
GiNaC::ptr< T >, 1069
parameter_set
GiNaC::fderivative, 644
paramset
GiNaC, 70
parent
GiNaC::class_info< OPT >, 405
parent_name
GiNaC::print_context_options, 1016
GiNaC::registered_class_options, 1081
parents_identified
GiNaC::class_info< OPT >, 405
partition
GiNaC::partition_generator, 957
GiNaC::partition_with_zero_parts_generator, 959
partition_generator
GiNaC::partition_generator, 956
partition_with_zero_parts_generator
GiNaC::partition_with_zero_parts_generator, 959
pathintegral_term
GiNaC::Gt_detail::pathintegral_term, 961
pathintegral_term_list
GiNaC::Gt_detail::pathintegral_term, 962
pattern_is_not_product
GiNaC::subs_options, 1169
pattern_is_product
GiNaC::subs_options, 1169
pderivative
GiNaC::function, 661
permutation_sign
GiNaC, 288
permute_free_index_to_front
GiNaC, 114
Pi
GiNaC, 293
PiEvalf
GiNaC, 252
pivot
GiNaC::matrix, 834
point
GiNaC::pseries, 1062
pointer
GiNaC::const_iterator, 444
GiNaC::const_postorder_iterator, 449
GiNaC::const_preorder_iterator, 452
pointer_to_map_function
GiNaC::pointer_to_map_function, 968
pointer_to_map_function_1arg
GiNaC::pointer_to_map_function_1arg< T1 >, 970
pointer_to_map_function_2args
GiNaC::pointer_to_map_function_2args< T1, T2 >, 972
pointer_to_map_function_3args

GiNaC::pointer_to_map_function_3args< T1, T2, T3 >, 975
pointer_to_member_to_map_function
GiNaC::pointer_to_member_to_map_function< C >, 978
pointer_to_member_to_map_function_1arg
GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >, 981
pointer_to_member_to_map_function_2args
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >, 984
pointer_to_member_to_map_function_3args
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >, 987
pole_error
GiNaC::pole_error, 989
polylog
GiNaC::Gt_detail::pathintegral_term, 963
polynomial
GiNaC::factor_options, 625
GiNaC::info_flags, 761
pop
GiNaC::Gt_detail::pathintegral_term_list, 965
pos_sig
GiNaC::minkmetric, 843
GiNaC::tensepsilon, 1215
posint
GiNaC::info_flags, 760
positive
GiNaC::domain, 516
GiNaC::info_flags, 760
possymbol
GiNaC::possymbol, 995
postorder_begin
GiNaC::ex, 569
postorder_end
GiNaC::ex, 569
pow
GiNaC, 253, 269, 270
GiNaC::matrix, 826
power
GiNaC::add, 336
GiNaC::function, 662
GiNaC::Gt_detail::pathintegral_term, 963
GiNaC::mul, 871
GiNaC::ncmul, 923
GiNaC::numeric, 942
GiNaC::power, 1001
power.cpp, 1352
power.h, 1353
power_const
GiNaC::pseries, 1060
power_dyn
GiNaC::numeric, 943
power_f
GiNaC::function_options, 702
power_func
GiNaC::function_options, 688–690, 695
power_funcp
GiNaC, 71
power_funcp_1
GiNaC, 72
power_funcp_10
GiNaC, 83
power_funcp_11
GiNaC, 85
power_funcp_12
GiNaC, 86
power_funcp_13
GiNaC, 88
power_funcp_14
GiNaC, 89
power_funcp_2
GiNaC, 73
power_funcp_3
GiNaC, 74
power_funcp_4
GiNaC, 75
power_funcp_5
GiNaC, 77
power_funcp_6
GiNaC, 78
power_funcp_7
GiNaC, 79
power_funcp_8
GiNaC, 81
power_funcp_9
GiNaC, 82
power_funcp_exvector
GiNaC, 90
power_use_exvector_args
GiNaC::function_options, 705
precedence
GiNaC::add, 327
GiNaC::basic, 366
GiNaC::clifford, 414
GiNaC::container< C >, 473
GiNaC::expairseq, 610
GiNaC::function, 656
GiNaC::Gt, 714
GiNaC::indexed, 752
GiNaC::integral, 767
GiNaC::mul, 860
GiNaC::ncmul, 918
GiNaC::numeric, 934
GiNaC::power, 1001
GiNaC::pseries, 1052
GiNaC::relational, 1088
GiNaC::structure< T, ComparisonPolicy >, 1137
prefactor
GiNaC::Gt_detail::pathintegral_term, 963
prem
GiNaC, 223
preorder_begin
GiNaC::ex, 568
preorder_end

GiNaC::ex, 569
prepend
GiNaC::container< C >, 478
prime
GiNaC::info_flags, 761
primitive_dirichlet_character
GiNaC, 205
primpart
GiNaC::ex, 584
print
GiNaC::_numeric_digits, 318
GiNaC::basic, 365
GiNaC::ex, 570
GiNaC::expair, 600
GiNaC::fderivative, 639
GiNaC::function, 656
GiNaC::structure< T, ComparisonPolicy >, 1136
print.cpp, 1354
print.h, 1354
GINAC_DECLARE_PRINT_CONTEXT, 1356
GINAC_DECLARE_PRINT_CONTEXT_BASE, 1356
GINAC_DECLARE_PRINT_CONTEXT_COMMON, 1356
GINAC_IMPLEMENT_PRINT_CONTEXT, 1356
print_add
GiNaC::add, 335
print_cache
GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1230
print_context
GiNaC::print_context, 1014
print_context_class_info
GiNaC, 92
print_context_options
GiNaC::print_context_options, 1016
print_csrc
GiNaC::print_csrc, 1018
print_csrc_cl_N
GiNaC::print_csrc_cl_N, 1020
print_csrc_double
GiNaC::print_csrc_double, 1022
print_csrc_float
GiNaC::print_csrc_float, 1024
print_dflt
GiNaC::print_dflt, 1026
print_dispatch
GiNaC::basic, 380
print_dispatch_table
GiNaC::function_options, 702
GiNaC::registered_class_options, 1081
print_elements
GiNaC::matrix, 834
print_func
GiNaC::function_options, 696–698
GiNaC::registered_class_options, 1080, 1081
print_func< print_context >
GiNaC, 135
print_func< print_dflt >
GiNaC, 98, 113, 282
print_funcp
GiNaC, 71
print_funcp_1
GiNaC, 72
print_funcp_10
GiNaC, 84
print_funcp_11
GiNaC, 85
print_funcp_12
GiNaC, 86
print_funcp_13
GiNaC, 88
print_funcp_14
GiNaC, 89
print_funcp_2
GiNaC, 73
print_funcp_3
GiNaC, 74
print_funcp_4
GiNaC, 76
print_funcp_5
GiNaC, 77
print_funcp_6
GiNaC, 78
print_funcp_7
GiNaC, 80
print_funcp_8
GiNaC, 81
print_funcp_9
GiNaC, 82
print_funcp_exvector
GiNaC, 91
print_functor
GiNaC::print_functor, 1027
print_index
GiNaC::idx, 739
print_index_dimensions
GiNaC::print_options, 1035
print_indexed
GiNaC::indexed, 757
print_integer_csrc
GiNaC, 235
print_latex
GiNaC::print_latex, 1031
print_memfun_handler
GiNaC::print_memfun_handler< T, C >, 1033
print_numeric
GiNaC::numeric, 952
print_operator
GiNaC, 273
print_overall_coeff
GiNaC::mul, 869
print_power
GiNaC::power, 1009
print_ptrfun_handler
GiNaC::print_ptrfun_handler< T, C >, 1036

print_python
GiNaC::print_python, 1039
print_python_repr
GiNaC::print_python_repr, 1040
print_real_cl_N
GiNaC, 237
print_real_csrc
GiNaC, 236
print_real_number
GiNaC, 235
print_series
GiNaC::pseries, 1060
print_sym_pow
GiNaC, 269
print_tree
GiNaC::print_tree, 1042
print_use_exvector_args
GiNaC::function_options, 705
printindices
GiNaC::indexed, 757
printpair
GiNaC::expairseq, 615
printraw
GiNaC::archive, 342
GiNaC::archive_node, 353
printseq
GiNaC::container< C >, 477
GiNaC::expairseq, 615
product_to_exvector
GiNaC, 140
property
GiNaC::archive_node::property, 1043
property_info
GiNaC::archive_node::property_info, 1045
property_type
GiNaC::archive_node, 348
propinfovector
GiNaC::archive_node, 347
props
GiNaC::archive_node, 354
pseries
GiNaC::pseries, 1051, 1052
pseries.cpp, 1357
pseries.h, 1358
psi
GiNaC, 165, 245
psi1_deriv
GiNaC, 174
psi1_eval
GiNaC, 174
psi1_evalf
GiNaC, 174
psi1_series
GiNaC, 174
psi2_deriv
GiNaC, 175
psi2_eval
GiNaC, 175
psi2_evalf
GiNaC, 174
psi2_series
GiNaC, 175
ptr
GiNaC::pointer_to_map_function, 968
GiNaC::pointer_to_map_function_1arg< T1 >, 970
GiNaC::pointer_to_map_function_2args< T1, T2 >, 973
GiNaC::pointer_to_map_function_3args< T1, T2, T3 >, 975
GiNaC::pointer_to_member_to_map_function< C >, 978
GiNaC::pointer_to_member_to_map_function_1arg< C, T1 >, 981
GiNaC::pointer_to_member_to_map_function_2args< C, T1, T2 >, 984
GiNaC::pointer_to_member_to_map_function_3args< C, T1, T2, T3 >, 987
GiNaC::ptr< T >, 1066
ptr.h, 1358
PTYPE_BOOL
GiNaC::archive_node, 348
PTYPE_NODE
GiNaC::archive_node, 348
PTYPE_STRING
GiNaC::archive_node, 348
PTYPE_UNSIGNED
GiNaC::archive_node, 348
purely_indefinite
GiNaC::status_flags, 1130
python
GiNaC, 267
python_repr
GiNaC, 267

q_expansion_modular_form
GiNaC::Eisenstein_h_kernel, 534
GiNaC::Eisenstein_kernel, 544
GiNaC::modular_form_kernel, 851
qExpand
GiNaC::Gt, 718
qExpand_minOrder
GiNaC::Gt, 723
qExpand_stepSize
GiNaC::Gt, 723
quo
GiNaC, 222

r
GiNaC::Eisenstein_h_kernel, 535
rank
GiNaC, 216, 217
GiNaC::matrix, 831
rational
GiNaC::info_flags, 760
rational_function
GiNaC::info_flags, 761

rational_polynomial
GiNaC::info_flags, 761
rbegin
GiNaC::container< C >, 480
read_archive
GiNaC::basic, 381
GiNaC::clifford, 415
GiNaC::color, 434
GiNaC::constant, 462
GiNaC::container< C >, 475
GiNaC::expairseq, 613
GiNaC::fderivative, 641
GiNaC::function, 660
GiNaC::Gt, 716
GiNaC::idx, 735
GiNaC::indexed, 754
GiNaC::integral, 770
GiNaC::matrix, 823
GiNaC::minkmetric, 842
GiNaC::numeric, 939
GiNaC::power, 1007
GiNaC::pseries, 1057
GiNaC::relational, 1089
GiNaC::spinidx, 1117
GiNaC::symbol, 1183
GiNaC::symmetry, 1195
GiNaC::tensepsilon, 1214
GiNaC::varidx, 1249
GiNaC::wildcard, 1257
read_real_float
GiNaC, 235
read_unsigned
GiNaC, 94
real
GiNaC, 257
GiNaC::domain, 516
GiNaC::info_flags, 760
GiNaC::numeric, 951
real_part
GiNaC, 120
GiNaC::add, 331
GiNaC::basic, 378
GiNaC::constant, 461
GiNaC::container< C >, 476
GiNaC::ex, 574
GiNaC::function, 659
GiNaC::indexed, 753
GiNaC::matrix, 823
GiNaC::mul, 863
GiNaC::ncmul, 921
GiNaC::numeric, 939
GiNaC::power, 1007
GiNaC::pseries, 1056
GiNaC::realsymbol, 1075
GiNaC::symbol, 1183
real_part_conjugate
GiNaC, 146
real_part_eval
GiNaC, 146
real_part_evalf
GiNaC, 146
real_part_expl_derivative
GiNaC, 147
real_part_f
GiNaC::function_options, 701
real_part_func
GiNaC::function_options, 678–680, 695
real_part_funcp
GiNaC, 70
real_part_funcp_1
GiNaC, 71
real_part_funcp_10
GiNaC, 83
real_part_funcp_11
GiNaC, 84
real_part_funcp_12
GiNaC, 86
real_part_funcp_13
GiNaC, 87
real_part_funcp_14
GiNaC, 89
real_part_funcp_2
GiNaC, 72
real_part_funcp_3
GiNaC, 74
real_part_funcp_4
GiNaC, 75
real_part_funcp_5
GiNaC, 76
real_part_funcp_6
GiNaC, 77
real_part_funcp_7
GiNaC, 79
real_part_funcp_8
GiNaC, 80
real_part_funcp_9
GiNaC, 81
real_part_funcp_exvector
GiNaC, 90
real_part_imag_part
GiNaC, 147
real_part_print_latex
GiNaC, 146
real_part_real_part
GiNaC, 147
real_part_use_exvector_args
GiNaC::function_options, 704
really_subs_idx
GiNaC::subs_options, 1169
realsymbol
GiNaC::realsymbol, 1074, 1075
recombine_pair_to_ex
GiNaC::add, 334
GiNaC::expairseq, 616
GiNaC::mul, 867
reduced_matrix

GiNaC, 214
reeval_ncmul
GiNaC, 219
GiNaC::ncmul, 923
refcount
GiNaC::refcounted, 1079
refcounted
GiNaC::refcounted, 1078
reference
GiNaC::const_iterator, 444
GiNaC::const_postorder_iterator, 450
GiNaC::const_preorder_iterator, 453
REGISTER_FUNCTION
function.h, 1305
GiNaC, 145, 147, 149, 151, 153, 154, 156–159, 161, 162, 167, 168, 171–173, 178, 179, 181, 184, 186, 188, 189, 191–193, 195–198, 200–203
register_new
GiNaC::function, 663
registered_class_info
GiNaC, 92
registered_class_options
GiNaC::registered_class_options, 1080
registered_functions
GiNaC::function, 662
registrar.cpp, 1359
registrar.h, 1359
GINAC_DECLARE_REGISTERED_CLASS, 1361
GINAC_DECLARE_REGISTERED_CLASS_COMMON, 1360
GINAC_DECLARE_REGISTERED_CLASS_NO_CTORS, 1360
GINAC_IMPLEMENT_REGISTERED_CLASS, 1361
GINAC_IMPLEMENT_REGISTERED_CLASS_OPT, 1361
GINAC_IMPLEMENT_REGISTERED_CLASS_OPT_T, 1362
regularise
GiNaC::Gt, 717
relation
GiNaC::info_flags, 761
relation_equal
GiNaC::info_flags, 761
relation_greater
GiNaC::info_flags, 761
relation_greater_or_equal
GiNaC::info_flags, 761
relation_less
GiNaC::info_flags, 761
relation_less_or_equal
GiNaC::info_flags, 761
relation_not_equal
GiNaC::info_flags, 761
relational
GiNaC::relational, 1088
relational.cpp, 1362
relational.h, 1363
relative_integration_error
GiNaC::integral, 772
rem
GiNaC, 222
remember
GiNaC::function_options, 699
remember.cpp, 1363
remember.h, 1364
remember_assoc_size
GiNaC::function_options, 703
remember_size
GiNaC::function_options, 703
remember_strategy
GiNaC::function_options, 703
GiNaC::remember_table, 1097
GiNaC::remember_table_list, 1102
remember_table
GiNaC::remember_table, 1096
remember_table_entry
GiNaC::function, 664
GiNaC::remember_table_entry, 1099
remember_table_list
GiNaC::remember_table_list, 1102
remember_tables
GiNaC::remember_table, 1096
remove_all
GiNaC::container< C >, 479
remove_dirac_ONE
GiNaC, 106
remove_first
GiNaC::container< C >, 478
remove_last
GiNaC::container< C >, 478
remove_reference
GiNaC::refcounted, 1078
rename_dummy_indices
GiNaC, 139
rename_dummy_indices_uniquely
GiNaC, 142, 143
rend
GiNaC::container< C >, 480
replace_contr_index
GiNaC::tensor, 1226
replace_dim
GiNaC::idx, 738
replace_with_symbol
GiNaC, 232
reposition_dummy_indices
GiNaC, 140
GiNaC::indexed, 759
representation_label
GiNaC::clifford, 419
GiNaC::color, 436
reserve
GiNaC::container_storage< C >, 483
rest
GiNaC::expair, 601

result
GiNaC::remember_table_entry, 1100
result_type
GiNaC::map_function, 813
resultant
GiNaC, 234
return_type
GiNaC::add, 332
GiNaC::basic, 377
GiNaC::clifford, 416
GiNaC::color, 436
GiNaC::ex, 590
GiNaC::expairseq, 614
GiNaC::fail, 630
GiNaC::function, 661
GiNaC::function_options, 703
GiNaC::indexed, 755
GiNaC::integral, 769
GiNaC::matrix, 824
GiNaC::minkmetric, 842
GiNaC::mul, 866
GiNaC::ncmul, 921
GiNaC::power, 1008
GiNaC::relational, 1090
GiNaC::structure< T, ComparisonPolicy >, 1145
GiNaC::su3d, 1152
GiNaC::su3f, 1158
GiNaC::tensdelta, 1207
GiNaC::tensepsilon, 1214
GiNaC::tensmetric, 1221
GiNaC::tensor, 1226
return_type_tinfo
GiNaC::add, 333
GiNaC::basic, 378
GiNaC::clifford, 416
GiNaC::color, 436
GiNaC::ex, 590
GiNaC::function, 661
GiNaC::function_options, 703
GiNaC::indexed, 755
GiNaC::integral, 770
GiNaC::mul, 866
GiNaC::ncmul, 921
GiNaC::power, 1008
GiNaC::relational, 1091
GiNaC::structure< T, ComparisonPolicy >, 1146
rh
GiNaC::relational, 1093
rhs
GiNaC, 127
GiNaC::ex, 573
GiNaC::relational, 1092
rl
GiNaC::return_type_t, 1104
root
GiNaC::archive::archived_ex, 358
rotate_left
GiNaC, 287
row
GiNaC::matrix, 835
rows
GiNaC, 215
GiNaC::matrix, 824

s
GiNaC::const_postorder_iterator, 451
GiNaC::const_preorder_iterator, 454
GiNaC::derivative_map_function, 488
GiNaC::Eisenstein_h_kernel, 535
GiNaC::print_context, 1014
GiNaC::symbolset, 1188
S_deriv
GiNaC, 179
S_eval
GiNaC, 178
S_evalf
GiNaC, 178
S_print_latex
GiNaC, 179
S_series
GiNaC, 178
safe_bool
GiNaC::relational, 1087
same_metric
GiNaC::clifford, 417
scalar_mul_indexed
GiNaC::basic, 376
GiNaC::matrix, 822
GiNaC::structure< T, ComparisonPolicy >, 1144
seq
GiNaC::container_storage< C >, 484
GiNaC::expairseq, 621
GiNaC::pseries, 1062
GiNaC::remember_table_entry, 1100
serial
GiNaC::constant, 465
GiNaC::function, 664
GiNaC::G2_SERIAL, 706
GiNaC::G3_SERIAL, 707
GiNaC::iterated_integral2_SERIAL, 783
GiNaC::iterated_integral3_SERIAL, 784
GiNaC::psi1_SERIAL, 1063
GiNaC::psi2_SERIAL, 1064
GiNaC::symbol, 1187
GiNaC::zeta1_SERIAL, 1259
GiNaC::zeta2_SERIAL, 1260
series
GiNaC, 124
GiNaC::add, 329
GiNaC::basic, 373
GiNaC::Eisenstein_h_kernel, 532
GiNaC::Eisenstein_kernel, 542
GiNaC::ex, 579
GiNaC::fderivative, 640
GiNaC::function, 658
GiNaC::integral, 771
GiNaC::integration_kernel, 778

GiNaC::modular_form_kernel, 849
GiNaC::mul, 863
GiNaC::power, 1005
GiNaC::pseries, 1054
GiNaC::structure< T, ComparisonPolicy >, 1142
GiNaC::symbol, 1181
series_coeff
GiNaC::integration_kernel, 781
series_coeff_impl
GiNaC::basic_log_kernel, 393
GiNaC::Ebar_kernel, 524
GiNaC::ELi_kernel, 554
GiNaC::integration_kernel, 780
GiNaC::Kronecker_dtau_kernel, 794
GiNaC::Kronecker_dz_kernel, 803
GiNaC::multiple_polylog_kernel, 909
series_f
GiNaC::function_options, 702
series_func
GiNaC::function_options, 690–692, 696
series_funcp
GiNaC, 71
series_funcp_1
GiNaC, 72
series_funcp_10
GiNaC, 83
series_funcp_11
GiNaC, 85
series_funcp_12
GiNaC, 86
series_funcp_13
GiNaC, 88
series_funcp_14
GiNaC, 89
series_funcp_2
GiNaC, 73
series_funcp_3
GiNaC, 74
series_funcp_4
GiNaC, 75
series_funcp_5
GiNaC, 77
series_funcp_6
GiNaC, 78
series_funcp_7
GiNaC, 79
series_funcp_8
GiNaC, 81
series_funcp_9
GiNaC, 82
series_funcp_exvector
GiNaC, 91
series_to_poly
GiNaC, 271
series_use_exvector_args
GiNaC::function_options, 705
series_vec
GiNaC::integration_kernel, 782
set
GiNaC::matrix, 827
set_cache_step
GiNaC::integration_kernel, 780
set_name
GiNaC::function_options, 672
GiNaC::symbol, 1185
set_print_context
GiNaC, 266
set_print_func
GiNaC, 272
GiNaC::function_options, 700
GiNaC::registered_class_options, 1081
set_print_options
GiNaC, 266
set_refcount
GiNaC::refcounted, 1078
set_return_type
GiNaC::function_options, 698
set_symmetry
GiNaC::function_options, 699
set_TeX_name
GiNaC::symbol, 1185
set_type
GiNaC::symmetry, 1196
setArgs
GiNaC::Gt, 714
setflag
GiNaC::basic, 384
shaker_sort
GiNaC, 288
share
GiNaC::ex, 593
shift_exponents
GiNaC::pseries, 1060
show_statistics
GiNaC::remember_table, 1096
simplify_indexed
GiNaC, 125, 141
GiNaC::ex, 586, 587
GiNaC::indexed, 758
simplify_indexed_product
GiNaC, 141
GiNaC::indexed, 758
sin
GiNaC, 238
sin_conjugate
GiNaC, 187
sin_deriv
GiNaC, 187
sin_eval
GiNaC, 187
sin_evalf
GiNaC, 186
sin_imag_part
GiNaC, 187
sin_real_part
GiNaC, 187

sinh
GiNaC, 241
sinh_conjugate
GiNaC, 197
sinh_deriv
GiNaC, 196
sinh_eval
GiNaC, 196
sinh_evalf
GiNaC, 196
sinh_imag_part
GiNaC, 197
sinh_real_part
GiNaC, 196
size
GiNaC::basic_multi_iterator< T >, 397
GiNaC::Gt_detail::pathintegral_term_list, 965
smod
GiNaC, 248
GiNaC::add, 330
GiNaC::basic, 375
GiNaC::ex, 585
GiNaC::mul, 864
GiNaC::numeric, 938
GiNaC::structure< T, ComparisonPolicy >, 1143
solve
GiNaC::matrix, 830
sort
GiNaC::container< C >, 479
sort_
GiNaC::container< C >, 477
spinidx
GiNaC::spinidx, 1116
spinor_metric
GiNaC, 284
split_ex_to_pair
GiNaC::add, 333
GiNaC::expairseq, 615
GiNaC::mul, 866
spm
GiNaC::scalar_products, 1108
spmap
GiNaC, 91
spmapkey
GiNaC::spmapkey, 1128
sprem
GiNaC, 224
sqrfree
GiNaC, 231
sqrfree_parfrac
GiNaC, 231
sqrfree_yun
GiNaC, 230
sqrt
GiNaC, 251, 270
sr_gcd
GiNaC, 226
STATISTICS
normal.cpp, 1341
std, 313
swap, 314
std::equal_to< GiNaC::ex >, 555
operator(), 555
std::hash< GiNaC::ex >, 727
operator(), 727
std::less< GiNaC::ptr< T > >, 806
operator(), 806
std::less< ptr< T > >
GiNaC::ptr< T >, 1068
step
GiNaC, 254
GiNaC::numeric, 944
step_conjugate
GiNaC, 152
step_eval
GiNaC, 152
step_evalf
GiNaC, 151
step_imag_part
GiNaC, 152
step_real_part
GiNaC, 152
step_series
GiNaC, 152
STLT
GiNaC::container< C >, 472
GiNaC::container_storage< C >, 482
store_remember_table
GiNaC::function, 662
struct_compare
GiNaC::compare_all_equal< T >, 437
GiNaC::compare_bitwise< T >, 438
GiNaC::compare_std_less< T >, 439
struct_is_equal
GiNaC::compare_all_equal< T >, 437
GiNaC::compare_bitwise< T >, 438
GiNaC::compare_std_less< T >, 439
structure
GiNaC::structure< T, ComparisonPolicy >, 1135
structure< T, CP >::reg_info
GiNaC, 296
structure.h, 1364
sub
GiNaC::matrix, 825
GiNaC::numeric, 941
sub_dyn
GiNaC::numeric, 942
sub_matrix
GiNaC, 214
subs
GiNaC, 127, 128
GiNaC::basic, 370
GiNaC::clifford, 418
GiNaC::container< C >, 474
GiNaC::ex, 575, 576
GiNaC::expairseq, 612

GiNaC::Gt, 715
GiNaC::idx, 735
GiNaC::matrix, 821
GiNaC::numeric, 936
GiNaC::power, 1005
GiNaC::pseries, 1054
GiNaC::relational, 1089
GiNaC::structure< T, ComparisonPolicy >, 1139
GiNaC::symbol, 1181
subs_algebraic
GiNaC::subs_options, 1169
subs_no_pattern
GiNaC::subs_options, 1169
subs_one_level
GiNaC::basic, 381
subschildren
GiNaC::container< C >, 481
GiNaC::expairseq, 621
subsvalue
GiNaC, 204
successful_hits
GiNaC::remember_table_entry, 1100
sufficiently_accurate
GiNaC::lanczos_coeffs, 805
suppress_branchcut
GiNaC::series_options, 1108
swap
GiNaC, 127, 132
GiNaC::ex, 568
GiNaC::expair, 600
GiNaC::ptr< T >, 1067
std, 314
swapped
GiNaC::sy_swap, 1171
sy_anti
GiNaC, 279, 280
sy_cycl
GiNaC, 280, 281
sy_is_less
GiNaC::sy_is_less, 1170
GiNaC::symmetry, 1197
sy_none
GiNaC, 278
sy_swap
GiNaC::sy_swap, 1171
GiNaC::symmetry, 1197
sy_symm
GiNaC, 279
sym
GiNaC::sym_desc, 1174
sym_desc
GiNaC::sym_desc, 1173
sym_desc_vec
GiNaC, 92
symbol
GiNaC::info_flags, 761
GiNaC::symbol, 1180
symbol.cpp, 1365
symbol.h, 1366
symbolic_matrix
GiNaC, 213, 217
symbolset
GiNaC::symbolset, 1188
symm
GiNaC, 277
GiNaC::terminfo, 1228
symmetric
GiNaC::symmetry, 1194
symmetric2
GiNaC, 275
symmetric3
GiNaC, 276
symmetric4
GiNaC, 276
symmetrize
GiNaC, 125, 277, 281
GiNaC::ex, 589
symmetrize_cyclic
GiNaC, 126, 277, 281
GiNaC::ex, 589, 590
symmetry
GiNaC::symmetry, 1194
symmetry.cpp, 1367
symmetry.h, 1368
symmetry_type
GiNaC::symmetry, 1194
symminfo
GiNaC::symminfo, 1200
symmterm
GiNaC::symminfo, 1200
symtree
GiNaC::function_options, 705
GiNaC::indexed, 759
synthesize_func
GiNaC, 68

table_size
GiNaC::remember_table, 1097
tan
GiNaC, 239
tan_conjugate
GiNaC, 191
tan_deriv
GiNaC, 190
tan_eval
GiNaC, 190
tan_evalf
GiNaC, 189
tan_imag_part
GiNaC, 190
tan_real_part
GiNaC, 190
tan_series
GiNaC, 190
tanh
GiNaC, 241
tanh_conjugate

GiNaC, 200
tanh_deriv
GiNaC, 199
tanh_eval
GiNaC, 199
tanh_evalf
GiNaC, 199
tanh_imag_part
GiNaC, 200
tanh_real_part
GiNaC, 199
tanh_series
GiNaC, 199
tau
GiNaC::Gt, 724
GiNaC::Kronecker_dz_kernel, 804
tauToFundamental
GiNaC::Gt, 717
tcoeff
GiNaC::ex, 578
tensepsilon
GiNaC::tensepsilon, 1213
tensor
GiNaC, 293
tensor.cpp, 1369
tensor.h, 1370
terminfo
GiNaC::terminfo, 1228
terms
GiNaC::Gt_detail::pathintegral_term_list, 966
test
GiNaC::has_distance< T >, 726
test_and_set_nparams
GiNaC::function_options, 700
TEST_PERMUTATION
color.cpp, 1275
TeX_name
GiNaC::constant, 465
GiNaC::function_options, 701
GiNaC::symbol, 1187
tgamma
GiNaC, 244, 245
tgamma_conjugate
GiNaC, 172
tgamma_deriv
GiNaC, 172
tgamma_eval
GiNaC, 171
tgamma_evalf
GiNaC, 171
tgamma_series
GiNaC, 172
thiscontainer
GiNaC::clifford, 416
GiNaC::color, 435
GiNaC::container< C >, 476, 477
GiNaC::fderivative, 640
GiNaC::function, 658
GiNaC::indexed, 754, 755
GiNaC::ncmul, 920
thisexpairseq
GiNaC::add, 333
GiNaC::expairseq, 614, 615
GiNaC::mul, 866
tinfo
GiNaC::return_type_t, 1104
tinfo_key
GiNaC::registered_class_options, 1081
to_cl_N
GiNaC::numeric, 951
to_double
GiNaC, 257
GiNaC::numeric, 950
to_int
GiNaC, 256
GiNaC::numeric, 950
to_integer
GiNaC::Gt, 722
to_long
GiNaC, 257
GiNaC::numeric, 950
to_numeric
GiNaC::Gt, 722
to_polynomial
GiNaC, 123
GiNaC::basic, 374
GiNaC::ex, 581
GiNaC::expairseq, 612
GiNaC::numeric, 937
GiNaC::power, 1006
GiNaC::structure< T, ComparisonPolicy >, 1143
GiNaC::symbol, 1182
to_rational
GiNaC, 123
GiNaC::basic, 374
GiNaC::ex, 580
GiNaC::expairseq, 611
GiNaC::numeric, 937
GiNaC::power, 1006
GiNaC::structure< T, ComparisonPolicy >, 1142
GiNaC::symbol, 1182
toggle_dot
GiNaC::spinidx, 1119
toggle_variance
GiNaC::varidx, 1251
toggle_variance_dot
GiNaC::spinidx, 1119
too_late
GiNaC::_numeric_digits, 319
trace
GiNaC, 216
GiNaC::matrix, 828
trace_string
GiNaC, 103
TransformExpressionWithCache

GiNaC::Gt_detail::TransformExpressionWithCache< Type >, 1230
transpose
GiNaC, 215
GiNaC::matrix, 827
traverse
GiNaC::ex, 577
traverse_postorder
GiNaC::ex, 577
traverse_preorder
GiNaC::ex, 576
tree
GiNaC, 267
tree_node
GiNaC::class_info< OPT >::tree_node, 1233
trig_info
GiNaC, 188
trivial
GiNaC::composition_generator, 441
tryfactsubs
GiNaC, 217
type
GiNaC::archive_node::property, 1044
GiNaC::archive_node::property_info, 1045
GiNaC::symmetry, 1198

uintvector
GiNaC, 91
unarch_map
GiNaC::unarchive_table_t, 1235
unarch_table_instance
GiNaC, 293
unarchive
GiNaC::archive_node, 352
unarchive_ex
GiNaC::archive, 339, 340
unarchive_map_t
GiNaC, 68
unarchive_table_t
GiNaC::unarchive_table_t, 1234
unatomize
GiNaC::archive, 342
unique
GiNaC::container< C >, 479
unique_
GiNaC::container< C >, 478, 481
unit
GiNaC::ex, 582
unit_matrix
GiNaC, 213, 217
unitcontprim
GiNaC::ex, 585
unlikely
compiler.h, 1278
unlink_ex
GiNaC, 132
unsignedvector
GiNaC, 91
USE_REMEMBER
normal.cpp, 1341
use_remember
GiNaC::function_options, 703
use_return_type
GiNaC::function_options, 703
USE_SAME_DEGREE_FACTOR
factor.cpp, 1290
use_sr_gcd
GiNaC::gcd_options, 708
USE_TRIAL_DIVISION
normal.cpp, 1341
usecount
GiNaC::unarchive_table_t, 1235
used_indices
GiNaC::make_flat_inserter, 810
user_defined_kernel
GiNaC::user_defined_kernel, 1241
uses_Laurent_series
GiNaC::Eisenstein_h_kernel, 533
GiNaC::Eisenstein_kernel, 544
GiNaC::integration_kernel, 779
GiNaC::modular_form_kernel, 850
GiNaC::user_defined_kernel, 1242
utils.cpp, 1371
utils.h, 1374
DEFAULT_COMPARE, 1375
DEFAULT_CTOR, 1375
DEFAULT_PRINT, 1375
DEFAULT_PRINT_LATEX, 1376
utils_multi_iterator.h, 1376

v
GiNaC::basic_multi_iterator< T >, 399
GiNaC::sy_is_less, 1170
GiNaC::sy_swap, 1171
v1
GiNaC::spmapkey, 1129
v2
GiNaC::spmapkey, 1129
v_internal
GiNaC::multi_iterator_shuffle< T >, 898
v_orig
GiNaC::multi_iterator_shuffle< T >, 899
validate
GiNaC::indexed, 758
GiNaC::symmetry, 1196
value
GiNaC::archive_node::property, 1044
GiNaC::composition_generator::coolmulti::element, 546
GiNaC::has_distance< T >, 726
GiNaC::idx, 740
GiNaC::numeric, 954
value_type
GiNaC::const_iterator, 444
GiNaC::const_postorder_iterator, 449
GiNaC::const_preorder_iterator, 452
var
GiNaC::pseries, 1062

varidx
GiNaC::varidx, 1249
version.h, 1378
GINAC_LT_AGE, 1379
GINAC_LT_CURRENT, 1378
GINAC_LT_REVISION, 1379
GINACLIB_ARCHIVE_AGE, 1379
GINACLIB_ARCHIVE_VERSION, 1379
GINACLIB_MAJOR_VERSION, 1378
GINACLIB_MICRO_VERSION, 1378
GINACLIB_MINOR_VERSION, 1378
GINACLIB_STR, 1379
GINACLIB_STR_HELPER, 1379
GINACLIB_VERSION, 1379
version_major
GiNaC, 296
version_micro
GiNaC, 297
version_minor
GiNaC, 296

wild
GiNaC, 292
wildcard
GiNaC::wildcard, 1256
wildcard.cpp, 1380
wildcard.h, 1380
write_real_float
GiNaC, 235
write_unsigned
GiNaC, 94

x
GiNaC::basic_partition_generator::mpartition2, 853
GiNaC::Ebar_kernel, 525
GiNaC::ELi_kernel, 555
GiNaC::integral, 772
GiNaC::user_defined_kernel, 1242

y
GiNaC::Ebar_kernel, 525
GiNaC::ELi_kernel, 555
yes_type
GiNaC::has_distance< T >, 725

z
GiNaC::Gt, 724
GiNaC::Kronecker_dtau_kernel, 795
GiNaC::multiple_polylog_kernel, 910
z_j
GiNaC::Kronecker_dz_kernel, 804
zeta
GiNaC, 163, 164, 244
zeta1_deriv
GiNaC, 181
zeta1_eval
GiNaC, 181
zeta1_evalf
GiNaC, 181
zeta1_print_latex
GiNaC, 181
zeta2_deriv
GiNaC, 182
zeta2_eval
GiNaC, 182
zeta2_evalf
GiNaC, 182
zeta2_print_latex
GiNaC, 182
zetaderiv_deriv
GiNaC, 158
zetaderiv_eval
GiNaC, 157
zi
GiNaC::Gt::kernel, 787
zisToFundamental
GiNaC::Gt, 717