A Perspective Of New Foundations For Knowledge Maintenance Systems: Research Program †

Alexei Yu. Muravitsky
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, New York 10012-1185
Internet: mrvtskya@acf4.nyu.edu

Abstract

We propose to provide new mathematical foundations for the design of knowledge-based systems. The underlying idea is that the knowledge which the computer ("artificial agent") operates with is considered as a kind of abstract data type. In this context, a relation of approximation arises in a natural way when one imagines the computer as operating in a changing information environment ("information flow"). This notion of approximation can be studied using the techniques that have been developed for domain theory in the context of denotational semantics of programming languages.

† **Acknowledgements**: The text below was first submitted as a proposal to the National Science Foundation in October 1994; then it was extended and partially changed. The first version was thoroughly edited by Martin Davis to whom the author is gratefully thankful. Any current mistakes, however, are my own. I am also cordially appreciative to Will Klump for correcting my English at different stages of my preparation. The initial impact for the outlined approach I received from reading Belnap's papers [Bel 75, Bel 76]. Nuel Belnap has constantly supported me in letters and E-mail messages. I am very grateful to him for that help.

1 Introduction

The main thrust of the proposed investigation is to provide new mathematical foundations for knowledge maintenance systems. The point of departure is to regard the states of knowledge of the system as elements of an *effectively presented domain*. The main purpose is to find and develop the notions of many-valued structures appropriate for the representation of formalized declarative knowledge. Many-valued logics based on such structures are meant to determine the epistemic states of a knowledge-based system. The search for suitable deductive systems will need to yield descriptions of suitable *Scott information systems* determining (up to isomorphism) the domains which result from the semantic description of epistemic states. Emphasis should be placed on research on computability and tractability of the relations arising from the approximation, and of the operations of the knowledge transformation, as well as of dependence of the domain structure on the structure of information flow.

A principal area to be studied is the characterization of continuous computable operations on the domains being considered, which are thought of simply as knowledge transformers by means of which the system maintains itself. Strategies correcting the computer's behavior such as backtracking can be studied cf. *e.g.* [Mur 94a]), though are left without attention in this paper. In Section 4, we propose the analysis of the possible evolution of the computer's knowledge by means of one or another epistemic logic.

The approach to be taken, on which considerable progress has already been made [KM 90, KM 93, Mur 93, Mur 94a, Mur 94b], involves combining two paradigms from different areas in computer science: Scott domains from denotational semantics and the use of non-classical logics to represent possibly inconsistent information. Although the intention in the proposed research will be to avoid concentrating exclusively on procedural definitions, as has tended to be the case in work on plan-recognition, contact with practice will be maintained by emphasis on the computability and tractability of relations and operations.

Our point of view is to be sharply distinguished from the anthropomorphic approach based on an attempt to directly model human behavior in computer systems. Although of course the modeling of the human mind remains one of the long term goals of artificial intelligence research, it may well be noted that in this field many techniques that originated in analytic philosophy are used, techniques that fundamentally depended on formalization in terms of abstract rather than anthropomorphic entities [Rus 18].

The knowledge representation process is taken to be a relation between information flow and an artificial agent, i.e., a computer based question-answering database system which responds to incoming information by appropriate changes in its data. The system is intended to be able to interact with changing information coming to it from a variety of sources. Consequently, it is appropriate to assume that the system can tolerate contradictory information in the form of an inconsistent input or as a side-effect of an "inoffensive" input. We consider the structure of information flow as represented by the propositions of a formal language, making assertions about the external world, labeled by values of an *epistemic structure*, symbolically $A : \tau$, and *constraints* that determine regularities taken into consideration, which we symbolically denote via $A_1 : \tau_1 \to A_2 : \tau_2$. Remaining within the framework of the linguistic interpretation of information knowledge, tracing at least to Frege (cf. *e.g.* [Fre 18]), and which is a point of the well-known *Knowledge Representation Hypothesis,* [Smi 82, Lev 86, Isr 93] the propositions are considered in the framework of a many-valued logic. As to constraints, they are realized in the knowledge representation process as *Scott-continuous computable* operations on the domain. In fact, such operations are the only ones permitted either for maintaining the state of its knowledge or corresponding to the acquisition of new data.

The underlying idea of the "information-knowledge" relation is that the computer acquires information from the outer word in the form of Scott-continuous operations on the domain of "real" and "ideal" epistemic states of the computer's knowledge and corrects it in accordance with imposed constraints that are also Scott-continuous operations on this domain. It is important that these operations be computable on the "real" elements of that domain — on its *effective basis.*

2 Representing Knowledge As a Domain

Recall (cf. [DB 90, GS 90] or [DSW 94, Chapter 16]) that a partially ordered set $\mathcal{P}$ is called *complete* if it has a bottom element $\perp$ and the least upper bound $\sqcup D$ exists for each directed subset $D \subseteq \mathcal{P}$. An element $x \in \mathcal{P}$ is said to be *compact*, if for any such directed subset $D \subseteq \mathcal{P}$,

$$x \leq \sqcup D \Rightarrow x \leq d \text{ for some } d \in D.$$

A complete partially ordered set is a *complete semilattice*, if each of its non-empty subsets has a greatest lower bound. And finally, a complete semilattice $\mathcal{P}$ is a *domain* if for each $x \in \mathcal{P}$,

$$x = \sqcup\{y \,|y \in \mathcal{P},\, y \text{ is compact}\}. \tag{1}$$

A complete partially ordered set $\mathcal{P}$ is called a *Scott domain*, if equation (1) is satisfied and if, in addition, the set

$$\{x \,|x \leq x_0, x \text{ is compact}\}$$

is directed for each element $x_0 \in \mathcal{P}$. Notice that every domain is also a Scott domain.

Given a partially ordered set $(\mathcal{P}, \leq)$ and $x \in \mathcal{P}$, let $\downarrow x \stackrel{\text{def}}{=} \{y \,|y \in \mathcal{P}, y \leq x\}$. Then, denote for any subset $u \subseteq \mathcal{P}$, $u \lhd \mathcal{P}$ as meaninng that for every $x \in \mathcal{P}$, the set $u \cap \downarrow x$ is a directed subset of $\mathcal{P}$. Finally, let $\mathcal{D}$ be a (Scott) domain and let $K(\mathcal{D})$ be the set of compact elements of $\mathcal{D}$. The domain $\mathcal{D}$ is said to be *effectively presented* if the relation $\leq$ on $K(\mathcal{D})$ is recursively decidable (with respect to some effective enumeration) and for every finite $u \subseteq K(\mathcal{D})$, it is effectively decidable whether $u \lhd K(\mathcal{D})$.

The importance of many-valued logics in computer science is generally recognized. They have been used in artificial intelligence [Gin 88], logic programming [Fit 85], algebraic specification of data types [Pig 90] and in other fields. In our work, we use the notion of *epistemic structure* to represent the (generally many-valued) truth values of propositions about the external world. This notion is in turn defined in terms of what we call *pre-epistemic structures.*

A *pre-epistemic structure* is an algebraic system $(\Im, \wedge, \vee, \neg, \boldsymbol{f}, \boldsymbol{t}, \sqsubseteq)$ satisfying the conditions:

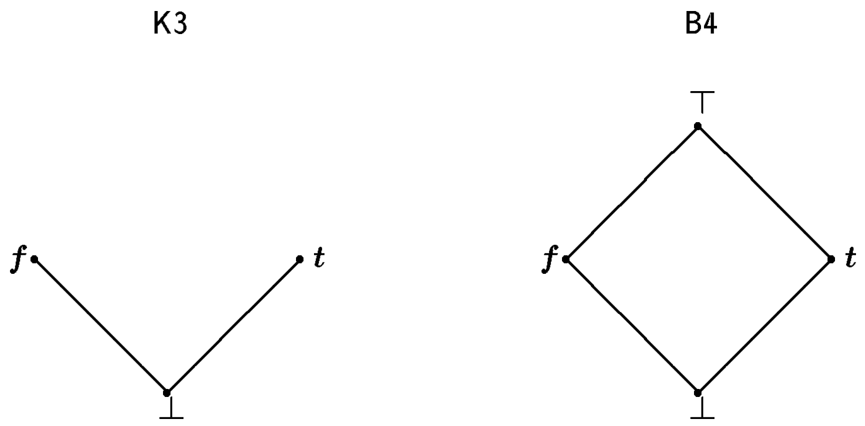
- $(\Im, \sqsubseteq)$ is a finite complete semilattice with respect to $\sqsubseteq$;

- $(\Im, \wedge, \vee)$ is a bounded lattice with: $\boldsymbol{f}, \boldsymbol{t} \in \Im$ and $\boldsymbol{f} \leq x \leq \boldsymbol{t}$ for every $x \in \Im$, where
  $x \leq y \iff x \wedge y = x$, or equivalently $x \vee y = y$;

- the operations $\wedge$, $\vee$ and $\neg$ are monotone on $\Im$ with respect to $\sqsubseteq$ with: $\neg \boldsymbol{f} = \boldsymbol{t}$ and $\neg \boldsymbol{t} = \boldsymbol{f}$.

Example 1: *Kleene's 3-valued logic* K3 and *Belnap's 4-valued bilattice* B4.

Figure 1: Pre-epistemic structures K3 and B4



Let us fix some pre-epistemic structure $\Im$. Let Var stand for the set of atomic assertions of a formal language. By a *setup*, we understand simply a mapping $s : \mathsf{Var} \rightarrow \Im$. We begin by considering a quite simple formal propositional language with connectives: $\wedge$ (conjunction), $\vee$ (disjunction) and $\neg$ (negation). Denote this language by L. Each setup can then be extended to the set of formulas of the language L as follows:

- $s(A \wedge B) = s\,(A) \wedge (B)$,
- $s(A \vee B) = s\,(A) \vee (B)$,
- $s(\neg A) = \neg s\,(A)$.

A setup s is called *finite* if the set

$$V(s) \stackrel{\text{def}}{=} \{\pi \,|\pi \in \mathsf{Var}, s(d\pi) \neq \bot\}$$

is finite.

Following [Bel 75, Bel 76], we introduce an order on the setups as follows:

$$s \leq s_1 \overset{\text{def}}{\iff} s(\pi) \sqsubseteq s_1(\pi) \text{ for every } \pi \in \mathsf{Var}.$$

This relation is obviously a partial order. We denote the partially ordered set of setups by Se. Notice that for the finite setups, the relation $\leq$ is recursively decidable.

Theorem 1 ([Mur 95a]) *The partially ordered set* Se *is a complete semi-lattice. Furthermore, every finite setup is a compact element of* Se*, that is, for any directed set D of setups,*

$$s \leq \sqcup D \Rightarrow s \leq s_1 \textit{ for some } s_1 \in D,$$

and conversely, every compact element in Se *is a finite setup. Moreover, for every setup s the equation*

$$s = \sqcup\{s' \,|\, s' \in \mathsf{Se},\ s' \textit{ is finite, and } s' \leq s\}$$

holds. Hence, Se *is an effectively presented domain.*

Following [Bel 75, Bel 76], we define an *epistemic state* to be any nonempty set of setups. Consider three orderings on the set of epistemic states:

- $\varepsilon_1 \leq \varepsilon_2$ iff for every $s_2 \in \varepsilon_2$ there is $s_1 \in \varepsilon_1$ such that $s_1 \leq s_2$;
- $\varepsilon_1 \preceq \varepsilon_2$ iff for every $s_1 \in \varepsilon_1$ there is $s_2 \in \varepsilon_2$ such that $s_1 \leq s_2$;
- $\varepsilon_1 \trianglelefteq \varepsilon_2$ iff $\varepsilon_1 \leq \varepsilon_2$ and $\varepsilon_1 \preceq \varepsilon_2$.

As easy to check, these orderings are pre-orders, that is, reflexive transitive relations.

An epistemic state is called *finite*, if the set

$$V(\varepsilon) \overset{\text{def}}{=} \cup\{V(s) \,|\, s \in \varepsilon\}$$

is finite, that is, if ε is a finite set of finite setups.

Now following [KM 93], let us write $m(\varepsilon)$ to stand for the minimal element of a finite state ε. Note that $m(\varepsilon)$ exists, is unique, and is itself a finite state, because of the Descending Chain Condition. It is easy to check that each $m(\varepsilon)$ consists of incomparable setups. We call a finite state ε *minimal* if $m(\varepsilon) = \varepsilon$. Let the partially ordered set of minimal states be denoted by ME. Let Se^* be the upper powerdomain of the domain Se in the sense of [GS 90].

Theorem 2 ([**Mur 95a**]) *Se^* is an effectively presented domain. The partially ordered set of compact elements of Se^* is isomorphic to ME. Moreover, for any $\varepsilon_1, \varepsilon_2 \in \mathsf{ME}$, $sup\{\varepsilon_1, \varepsilon_2\} \in \mathsf{ME}$, provided that $sup\{\varepsilon_1, \varepsilon_2\}$ exists.*

Example 2: In the case of Belnap's bilattice $\mathsf{B4}$, ME can be proved to be a distributive lattice [KM 93, Mur 94a]. On the other hand, in the case of $\mathsf{K3}$, if setups s_1 and s_2 are such that

$$s_1(\pi) = \begin{cases} \boldsymbol{t} & \text{for } \pi = p_1 \\ \bot & \text{otherwise} \end{cases} \quad \text{and} \quad s_2(\pi) = \begin{cases} \boldsymbol{f} & \text{for } \pi = p_1 \\ \bot & \text{otherwise,} \end{cases}$$

then $sup\{\{s_1\}, \{s_2\}\}$ does not exist.

The structure of the domain Se^* with ME as the compact elements is still rather complicated to work with, but it appears to be a good point of reference. We would like to have an observable structure so that it is easy to determine appropriate Scott-continuous operations to serve as knowledge transformers of the intelligent system. One way of doing this is to impose stronger conditions on pre-epistemic structures. A pre-epistemic structure is called an *epistemic structure* if it additionally satisfies:

- $x \sqsubseteq y \iff (\boldsymbol{f} \not\sqsubseteq x \text{ and } \boldsymbol{t} \not\sqsubseteq x) \text{ or } (\boldsymbol{t} \not\sqsubseteq x \text{ and } y \not\sqsubseteq \boldsymbol{t}) \text{ or }$
  $(\boldsymbol{f} \not\sqsubseteq x \text{ and } y \not\sqsubseteq \boldsymbol{f}) \text{ or } (y \not\sqsubseteq \boldsymbol{t} \text{ and } y \not\sqsubseteq \boldsymbol{f})$;
- $\{x \,|\, x \in \Im, \boldsymbol{f} \sqsubseteq x\}$ and $\{x \,|\, x \in \Im, x \sqsubseteq \boldsymbol{t}\}$ are $\vee$-semilattices.

Note that $\mathsf{K3}$ and $\mathsf{B4}$ both satisfy these additional conditions so they are epistemic structures.

Following [Bel 75, Bel 76], we define:

$$\varepsilon(A) \stackrel{\text{def}}{=} \sqcap\{s(A) \,|\, s \in \varepsilon\}$$

for every epistemic state ε and any formula A of L. Two epistemic states ε_1 and ε_2 are said to be *formula indistiguishable* if $\varepsilon_1(A) = \varepsilon_2(A)$ for every formula A.

Theorem 3 ([**Mur 95a**]) *Given two minimal states with respect to a fixed epistemic structure are formula indistinguishable if and only if they are equal.*

Theorem 3 seems to take a step towards a deductive characterization of minimal states. For Belnap's case (i.e. $\Im$=$\mathsf{B4}$), this was done in [Mur 94a] (Cf. Lemmas 10 and 12 there). It happened, in turn, to yield in [Mur 94b] a direct description of the domain Se^* via the notion of *Scott information system* [Sco 82, DB 90]. That description was based on the first degree entailment E_{fde} from [AB 75].

3 Knowledge Transformers

Having obtained an appropriate space for knowledge representation, we now must consider knowledge transformers. It is natural to take for these partially recursive functions of type $\mathsf{ME} \rightarrow \mathsf{ME}$. However, in order to take into account the domain structure of Se^* and the role of ME in Se^*, we propose to first investigate computable continuous operations on Se^* coordinated with ME in the sense to be explained. An operation $F : \mathsf{Se}^* \rightarrow \mathsf{Se}^*$ is called *Scott-continuous*, [Sco 72, GHKLMS 80] if for every directed $\{x_i \,|\, i \in I\}$ and $x \in \mathsf{Se}^*$,

$$x = \sqcup\{x_i \,|\, i \in I\} \text{ implies } F(x) = \sqcup\{F(x_i) \,|\, i \in I\}.$$

We say that F is *coordinated with* ME, if F is closed on ME, that is, $F(x) \in \mathsf{ME}$ whenever $x \in \mathsf{ME}$ [KM 93]. Finally, we define the set of pairs $G_F \subseteq \mathsf{ME} \times \mathsf{ME}$ as follows:

$$G_F \stackrel{\text{def}}{=} \{(\varepsilon_1, \varepsilon_2) \,|\, \varepsilon_2 \leq F(\varepsilon_1)\}.$$

The operation F is then called *computable*, if the corresponding relation G_F is recursively enumerable (with respect to a suitable effective enumeration) [GS 90]. Also, to make precise the natural stipulation that incoming information never decrease the computer's knowledge, we call the operation F *ampliative*, if $x \leq F(x)$ for every $x \in \mathsf{Se}^*$ [Bel 75].

To begin with, we consider for each formula A and element $\tau \in \Im$, the operation $[A:\tau]$ ($[A:\tau]$-*action*), defined as follows:

$$[A:\tau](\varepsilon) \stackrel{\text{def}}{=} \varepsilon \sqcup m(\{s \,|\, s \in \mathsf{Se}, \tau \sqsubseteq s(A), V(s) \subseteq V(A)\}),$$

where $V(A) \stackrel{\text{def}}{=} \{\pi \,|\, \pi \in \mathsf{Var} \text{ and } \pi \text{ occurs in } A\}$. Notice that if $\varepsilon \leq \varepsilon_1$ and $[A:\tau](\varepsilon_1)$ is defined, then, because Se^* is a domain (cf. Theorem 2), it must be a complete semilattice, so that $[A:\tau](\varepsilon)$ is defined as well. The intutive meaning is to inform the computer that the assertion expressed by the proposition A has a truth value at least τ. Receiving information, the computer changes its current state of knowledge in a "minimal" way.

Although this definition is certainly analoguous to the corresponding definition in Belnap's treatment (cf. [KM 93] and Theorem 4 in [Mur 94a]), they are not the same. The question arises here: How can we extend $[A:\tau]$ defined in this manner to the entire domain Se^* and then prove that this extended operation is continuous. We can attempt to proceed as follows: we define for every $x \in \mathsf{Se}^*$,

$$[A:\tau](x) \simeq \sqcup\{[A:\tau](\varepsilon)\,|\,\varepsilon \in \mathsf{ME}, \varepsilon \leq x\},$$

where $\simeq$ is Kleene's identity symbol for partially defined functions from [Kle 52]. The first questions arising here are to find conditions when $[A:\tau]$ is defined in x and continuous in this point. In the last case, we will have to refine the definition of continuity above to be suitable for the partially defined functions.

We may well hope to be able to give a characterization of all continuous ampliative functions coordinated with ME via $[A:\tau]$-actions in the same way as was done for the case of Belnap's 4-valued bilattice in [Mur 94b]. It was established there that every continuous ampliative operation coordinated with ME (with respect to $\mathsf{B4}$) is determined in terms of $[A]$-actions and a classification of such operations was proposed. Notice, however, that in the present case, we are dealing in general with partial operations.

Now let us turn to constraints. Recall that the constraints in this approach are to be certain continuous operations on Se^* that are computable on ME. Thus, their definitions depend on the prior development of the theory of such operations. On the other hand, constraints have to reflect regularities of the external world. Their definitions on Se^*, must therefore be intutively acceptable. We could begin by defining a constraint of the type $A_1:\tau_1 \rightarrow A_2:\tau_2$ to be a partial operation on ME by analogy with Belnap's treatment for the operation $[A \rightarrow B]$ (cf. Theorem 4 in [Mur 94a]). However, the proof of the Theorem 4 in [Mur 94a] made essential use of the fact that ME (with respect to epistemic structure $\mathsf{B4}$) is a distributive lattice.

4 An Epistemic Logic

Now we wish to consider what a computer can know about its own state of knowledge and how this state can evolve. We suppose that an intelligent system is formed to include facilities for knowledge revision in the form of continuous operations on the space of Se^*, which are coordinated with ME, and of some epistemic logic as the computer's knowledge of its epistemological capacity. The knowledge being embodied in such an intelligent system must be effectively accessible. In our case, it can be expressed in an epistemic language by means of *epistemic formulas* (e-formulas, for short) that are built up from atomic e-formulas of the form $(A : \tau)$, where $\tau \in \Im$, using ordinary propositional connectives $\wedge$, $\vee$, $\neg$ and modality $\Diamond$.

Thus, we define $\mathcal{A}$ as being an e-formula whenever $\mathcal{A}$ is an atomic e-formula or of the form $(\mathcal{B} \wedge \mathcal{C})$, $(\mathcal{B} \vee \mathcal{C})$, $\neg\mathcal{B}$ or $\Diamond\mathcal{B}$, where $\mathcal{B}$ and $\mathcal{C}$ are e-formulas.

Let $\mathcal{F}$ be a class of computable (or continuous, or continuous and ampliative) operations on (or, respectively, coordinated with) ME. We say that the minimal state ε_1 is *accessible* from ε_0 (symbolically, $\Re\varepsilon_0\varepsilon_1$), if there exists an operation $F \in \mathcal{F}$ such that $F(\varepsilon_0) = \varepsilon_1$. (cf. [Mur 93]). It might seem that is necessary to introduce the transitive closure of $\Re$. However, since the class $\mathcal{F}$ is closed under the composition of a finite number of operations, $\Re$ is transitive.

Now, we define the notion of *validity* of an e-formula in a minimal epistemic state (symbolically, $\varepsilon \models \mathcal{A}$) as follows:

$\varepsilon \models (A : \tau)$ iff $\varepsilon(A) = \tau$, where $\tau \in \Im$;
$\varepsilon \models (\mathcal{B} \wedge \mathcal{C})$ iff $\varepsilon \models \mathcal{B}$ and $\varepsilon \models \mathcal{C}$;
$\varepsilon \models (\mathcal{B} \vee \mathcal{C})$ iff $\varepsilon \models \mathcal{B}$ or $\varepsilon \models \mathcal{C}$;
$\varepsilon \models \neg\mathcal{B}$ iff not $\varepsilon \models \mathcal{B}$;
$\varepsilon \models \Diamond\mathcal{B}$ iff there is a state ε_0 such that $\Re\varepsilon\varepsilon_0$ and $\varepsilon_0 \models \mathcal{B}$.

We think of $\varepsilon \models (A : \tau)$ as meaning that the computer knows that an assignment of the formula A takes the value τ at the state ε. Then, we can consider three kinds of epistemic logic with respect to the class $\mathcal{F}$ and epistemic structure $\mathfrak{S}$.

Let $\mathsf{S}(\mathcal{F}, \mathfrak{S})$ be the set of all the e-formulas valid in every minimal epistemic state. We may call $\mathsf{S}(\mathcal{F}, \mathfrak{S})$ a logic, because it is not empty and is closed under *modus ponens*, although the operation of substitution is not defined for it. Then, let $\mathsf{S}_0(\mathcal{F}, \mathfrak{S})$ be the set of formulas of the purely modal propositional language such that each formula that results on substituting atomic epistemic formulas for propositional variables is a valid formula of $\mathsf{S}(\mathcal{F},\mathfrak{S})$. Finally, let $\mathsf{S}_1(\mathcal{F},\mathfrak{S})$ be the set of modal formulas, each substitution instance of which belongs to $\mathsf{S}_0(\mathcal{F},\mathfrak{S})$. We see that, by definition, this last logic is closed under substitution. Note that for the class $\mathcal{F}$ of continuous ampliative operations coordinated with ME, the logic $\mathsf{S}(\mathcal{F},\mathsf{B4})$ has been proved to be decidable (cf. [Mur 93]).

5 Implementation and Further Research

Although knowledge-based system field has increasingly added value in industrial and commercial settings during the two past decades, "the difficulties associated with market and technology development have created a widespread impression that the [existing] technology somehow failed" [HJ 94]. This feeling of failure has been recently expressed in the proceedings of *International Conference on Principles of Knowledge Representation and Reasoning* as follows:

> "We concede that a large of KR research probably does not have any immediate impact on building Artificial Intelligence systems" [LN 94].

Nonetheless, we aim to ground the approach described in the previous sections for designing experimental knowledge-processing systems. We find the courage in our recent developments concerning both investigation of domains based on the simplest epistemic structures of Kleene's and Belnap's logics in [KM 93, Mur 93, Mur 94a, Mur 94b, Mur 95b] and a more general consideration in [Mur 95a] where the computational aspect is stressed. Moreover, building and evaluating real experimental systems is the only way to understand the problems of knowledge maintenance efficiency in the framework of the new proposed techniques.

The starting point is to choose an appropriate epistemic structure $\mathfrak{S}$ generating the corresponding effectively presented domain Se^*. Thus, the computer-represented knowledge is seen to be realized in the domain's compact elements, ME, with which each element in Se^* can be approached with any exactness which Scott topology admits. The elements of ME, in turn, being *objects* to deploy knowledge, can be represented *semantically*, that is, in the form of epistemic states, or *deductively*, in the form of finite elements of relevant information systems. Notice that, though semantic representation is often more intuitively transparent, the deductive representation of domain, and especially its compact elements, facilitates investigation of operations on the domain, considered as appropriate knowledge transformers. It is more comfortable to work with formulas rather than with a set (even if it is finite) of setups (that is, an epistemic state). All the more so because the relation $\leq$ on the epistemic states is replaced by set inclusion $\subseteq$ on sets of formulas. Some problems of not only theoretical interest arise here concerning finding an effective prover for a corresponding system of inference engine, for example, in connection with decidabilty of equality of two $[A:\tau]$-operations.

The principal goal is to investigate computable operations on ME and, to begin with, continuous and ampliative operations on Se^* coordinated with ME understood to be knowledge transfomers for different natural epistemic structures. For implementation purposes, the characterization of all continuous computable ampliative operations, received in [Mur 94b] for the case $\mathfrak{S} = \mathsf{B4}$, can be used. We also aim to consider, along with Se^*, the cases of the *lower* and *convex powerdomains* in the sense of [GS 90], that is, the powerdomain constructions with respect to the relations $\preceq$ and $\trianglelefteq$ as defined in the Section 2, respectively.

Concerning epistemic logics as defined in the Section 3, the problems are to find tractable algorithms for their decidability. The question of their complexity may turn out to be quite difficult even for particular epistemic structures. Though for the case $\mathfrak{S} = \mathsf{B4}$, the epistemic logic can be installed into a knowledge-processing system aiming to tolerate contradictory information.

Finally, an important problem is to go beyond the propositional language or to consider richer propositional languages. The second alternative leads to the enrichment of the notion of epistemic structure with all the new problems arising from this. The first poses the problem of how to handle with the computer's finite resources potentially infinite information in a changing environment. (This problem was clearly recognized by Nuel Belnap; cf. [Bel 75, Bel 76].)

Indeed, assume that we have a constraint $\forall x Q(x) : \boldsymbol{t} \to P(a) : \boldsymbol{t}$ and that the computer receives the message that $Q(a) : \boldsymbol{t}$, and a is its only individual symbol. Then the computer must conclude that $P(a) : \boldsymbol{t}$. If we become aware subsequently that $Q(b) : \boldsymbol{f}$, we certainly cannot apply that constraint any longer. However, our information that $P(a) : \boldsymbol{t}$ has become questionable as well. Notice that this criticism does not refer to a constraint $\forall x(Q(x) : \boldsymbol{t} \rightarrow P(a) : \boldsymbol{t})$. Under this constraint, after two messages $Q(a) : \boldsymbol{t}$ and $Q(b) : \boldsymbol{f}$ are received by the computer, we have $P(a) : \boldsymbol{t}$, and we have no reason to doubt it. Thus, a starting point here is to investigate constraints of the form of *Horn sentences.* Note, by the way, that we thus accept a structure of information flow that does not satisfy some of the rules of inference of classical first-order logic.

References

[AB 75] A.R.Anderson and N.D.Belnap, Jr., *Entailment: the Logic of Relevance and Necessity*, vol. 1, Princeton, Princeton University Press, 1975.

[ABD 92] A.R.Anderson, N.D.Belnap, Jr., and J.M.Dunn, *Entailment: the Logic of Relevance and Necessity*, vol. 2, Princeton, Princeton University Press, 1992.

[Bar 92] J.Barwise, Constraints, Channels, and the Flow of Information, in: P.Aczel, D.Israel, Y.Katagiri, and S.Peters (eds.), *Situation Theory and Its Applications, Valuem 3*, CSLI Lecture Notes, no. 37, Stanford, California, 1993, pp. 3–27; also available as **Barwise-STA3.dvi** by public “ftp” from phil.indiana.edu, directory: pub .

[Bel 75] N.D.Belnap, A Useful Four-Valued Logic, in: J.M.Dunn and G.Epstein (eds.), Modern Uses of Multiple-Valued Logic, *Proceedings of International Symposium on Multiple-Valued Logic, 5th, Indiana University*, D. Reidel Publ. Co., 1975, pp. 9–37; also see §81 in [ABD 92].

[Bel 76] N.D.Belnap, How a Computer Should Think, in: G.Ryle (ed.), Contemporary Aspects of Philosophy, *Proceedings of the Oxford International Symposium*, 1975, Oriel Press, 1976, pp. 30–56; also see §81 in [ABD 92].

[DB 90] B.A.Davey and H.A.Priestley, *Introduction to Lattices and Order*, Cambridge, Cambridge University Press, 1990.

[DSW 94] M.D.Davis, R.Sigal and E.J. Weyuker, *Computability, Complexity, and Languages*, second edition, Boston and New York, Academic Press 1994.

[Fre 18] G.Frege, Der Gedanke: eine logische Untersuchung, *Logische Untersuchungen*, 1918; reprinted in: Vandenhoeck & Ruprecht in Göttingen, 1966, ss. 30–53.

[Fit 85] M.Fitting, A Kripke-Kleene Semantics for General Logic Programs, *Logic Programming*, 2(1985), pp. 295 –312.

[GHKLMS 80] G.Gierz, K.H.Hofmann, K.Keimel, J.D.Lawson, M.Mislove and D.S.Scott, *A Compendium of Continuous Lattices*, Springer-Verlag, 1980.

[Gin 88] M.I.Ginsberg, Multivalued Logics: A Uniform Approach to Inference in Artificial Intelligence, *Computational Intelligence*, 4(1988), no. 3, pp. 265–316.

[HJ 94] F.Hayes-Roth and N.Jacobstein, The State of Knoledge-Based Systems, *Communications of the ACM*, 37 (1994), no. 3, pp. 27–39.

[GS 90] C.A.Gunter and D.S.Scott, Semantic Domains, in: J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, Volume B: "Formal Models and Semantics", 1990, Elsevier, pp. 635–674.

[Isr 93] D.J.Israel, The Role(s) of Logic in Artificial Intelligence, in: D.M.Gabbay, C.J.Hogger and J.A.Robinson (eds.), *Handbook of Logic in Computer Science*, vol.1 , 1993, Oxford University Press, pp. 1–29.

[KM 90] Y.M.Kaluzhny and A.Yu.Muravitsky, Modification of Epistemic States, *10th Soviet Conference on Mathematical Logic*, Abstracts, 1990, Alma-Ata, p.73 (Russian).

[KM 93] Y.M.Kaluzhny and A.Yu.Muravitsky, A Knowledge Representation Based on the Belnap's Four-Valued Logic, *Journal of Applied Non-Classical Logics*, vol. 3, no. 2, 1993, pp. 189–203; also available via ftp from "cs.nyu.edu" in directory "/pub/local/mrvtskya" as file **knowledge.ps** .

[Kle 52] S.C.Kleene, *Introduction to Metamathematics*, P. Noordhof N.V., Groningen, 1952.

[LN 94] G.Lakemeyer and B.Nebel, Foundations of Knowledge Representation and Reasoning, in: G.Lakemeyer and B.Nebel (eds.), *Foundations of Knowledge Representation and Reasoning*, Lecture Notes in Artificial Intelligence, vol. 810, Springer-Verlag, 1994.

[Lev 86] H.J.Levesque, Knowledge Representation and Reasoning, *Annual Review of Computer Science*, vol. 1, 1986, pp. 255–287.

[Mur 93] A.Yu.Muravitsky, Logic of Information Knowledge, submitted to *Journal of Applied Non-Classical Logics*, 1993, available via ftp as a Courant Inst. Tech. Rep. from "cs.nyu.edu" in directory "pub/local/mrvtskya" as file **inform.ps**.

[Mur 94a] A.Yu.Muravitsky, A Framework for Knowledge-Based Systems, *Journal of Applied Non-Classical Logics*, 1994, (to appear); available via ftp as a Courant Inst. Tech. Rep. from "cs.nyu.edu" in directory "pub/local/mrvtskya" as file **frame.ps**.

[Mur 94b] A.Yu.Muravitsky, Knowledge Representation as Domains, submitted to *Annals of Mathematics and Artificial Intelligence*, 1994; available via ftp as a Courant Inst. Tech. Rep. from "cs.nyu.edu" in directory "pub/local/mrvtskya" as file **domain.ps**.

[Mur 95a] A.Yu.Muravitsky, Foundations for Knowledge Maintenance Systems, I, to be published.

[Mur 95b] A.Yu.Muravitsky, Knowledge Representation via Multivalued Logics, in: *The 2nd Workshop on Non-Standard Logic and Logical Aspects of Computer Science: Irkutsk, Russia, June 15–18, 1995* (*NSL' 95*) (to appear); available via ftp from "cs.nyu.edu" in directory "pub/local/mrvtskya" as file **multivalued.ps**.

[Pig 90] D.Pigozzi, Data Types over Multiple-Valued Logics, *Theoretical Computer Science*, 77(1990), pp. 161–194.

[Rus 18] B.Russell, The Philosophy of Logical Atomism, 1918. Reprinted in: R.C.Marsh (ed.), *Logic and Knowledge*, George Allen and Unwin, London, 1956.

[Sco 72] D.S.Scott, Continuous Lattices, in: *Lecture Notes in Mathematics*, vol. 274, Springer-Verlag, 1972, pp. 97–136.

[Sco 82] D.S.Scott, Domains for Denotational Semantics, in: *Lecture Notes in Computer Science*, no. 140, 1982, pp. 577–613.

[Smi 82] B.C.Smith, Reflection and Semantics in a Procedural Language, *Technical Report MIT/LCS/TR-272*, Massachusetts Institute of Technology, Laboratory of Computer Science, Cambridge, MA, 1982.