Some Knowledge Transformers: Infons and Constraints

Alexei Yu. Muravitsky
Department of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, New York 10012
Internet: `mrvtskya@acf4.nyu.edu`

April 12, 1995

The goal of this paper is twofold. First, it is to present a general scheme within which information is supposed to turn into the computer-represented knowledge and, second, to define two natural kinds of transfomers of this knowledge which this scheme thrusts us into considering.

We start with the question: What kind of relationship takes place in presenting knowledge into the computer? The word "knowledge" probably sounds somewhat ambiguous and "information" would be better. However, information comes as the computer's input, becoming the computer's knowledge (belief or experience) in the computer's memory. Thus, on the one hand, we have information outside of the computer as being the content of a piece of information flow. On the other hand, we see this information represented in the computer as its knowledge. What turns out information into the computer's knowledge? And if a piece of information is a message about a fact, then what makes a description of the fact informative?

The relation of naming, occurring in natural language proposes some hints, since the language can be regarded as reflecting the outer world. Although, Frege's description of the relation of naming which can be pictured by the scheme:

$$\boxed{\text{WORD}} \longrightarrow\!\triangleright \boxed{\text{SENSE}} \longrightarrow\!\triangleright \boxed{\text{OBJECT}}$$

will hardly do for characterizing the representation process. First of all, we realize in doing it the relation of "OBJECT–WORD" rather than the inverse one. Nevertheless, the Frege's scheme is helpful. We ask the question: What does "sense" mean in it? We think that it is not just "a third realm, a realm neither of ideas nor of worldly events" [BP 83], but a function[1], probably, partial, because some words may not have any references. Therefore, we prefer to rewrite the Frege's scheme as follows:

$$\boxed{\text{WORD}} \xrightarrow{\text{sense}}\!\triangleright \boxed{\text{OBJECT}}$$

And for our purposes in clarifying the process of knowledge representation, we will set out from the following functional scheme:

$$\boxed{\text{OBJECT}} \xrightarrow{\text{presentation}}\!\triangleright \boxed{\text{WORD}} \tag{1}$$

Had the last scheme been satisfied, it would have been suitable only for the representation of references of proper names. However, proper names only carry information insofar as they occur within propositions. [2] According to Freger's approach, the reference of a declarative proposition is its truth value. What forces the scheme (1) to work in the case of a proposition? The proposition is not informative until we have taken into account its *content*, i.e its "linguistic meaning" [BP 83]. What is the content of a proposition? Is it possible to talk about the content in a formal language not making it explicable via other notions?

Bertrand Russell says in [Rus 18] that propositions are used with connection with facts; they do not name facts, but facts confirm or disprove propositions. Thus, we arrive at the scheme:

$$\boxed{\text{FACT}} \xrightarrow[\text{disproves}]{\text{confirms}}\!\triangleright \boxed{\text{PROPOSITION}} \tag{2}$$

It is this connection that makes the proposition informative or uninformative.

[1] Frege says: "...also what I should like to call the *sense* of the sign, wherein the mode of presentation is contained" [Fre 892].

[2] A parallel suggests itself with Russell's denoting phrases: "...denoting phrases never have any meaning in themselves, but ... every proposition in whose verbal expression they occur has a meaning" [Rus 05].

To admit the thought containing in a proposition, i.e. its content, in other words, to make the proposition informative, "the step from the level of thoughts to the level of reference", [Fre 892] that is to say, from content to truth value has to be taken. [3] We do it every time when we are saying about proposition A "it is true that A", symbolically $A : \boldsymbol{t}$, or "it is false that A", symbolically $A : \boldsymbol{f}$, or else "it is unknown that A", symbolically $A : \bot$, as meaning in the last case that we do not know about the truth value of proposition A. On the other hand, placing the proposition A into an indirect context we deal with its content taking into account some unnamed facts that either confirm A or disprove it or else say nothing at all about it.

Recall that we are interested in the informativity of propositions not as a property of natural language, but a property of knowledge processing. Therefore, though, as Frege noted, the propositions A and "it is true that A" have the same content, provided A is really true, the computer appears to have much more to do with $A : \boldsymbol{t}$ than with A. For the sake of simplicity, we leave with no attention here to the analysis of information flow, being attempted in length in situation theory (see [BP 83, Bar 92, Dev 91]) where the *efficiency of language* is taken into consideration. We formalize below this notion in the mode of many-valued logic. In comparison with the situational approach, we consider ourselves as being more interested in investigation of the computer's strategies with respect to the structure of information flow. While the situational approach studies the logical structure of information facts, we would like to understand how this structure is reflected on the computer's activity in knowledge representation processing. Although, we begin with quite simplified notion of information flow.

First of all, in order for a piece of information to be accessible to processing by the computer, it must be a datum of some type. As we are limited by nature itself in data processing with computable operations, i.e. with the class of functions to which it is possible, in virtue of Church's Thesis, to give an precise mathematical sense, we appear to be faced here with some kind of structural limitations to be a data type, that is, in particular with some kind of an arrangement of the states of knowledge that may be involved. Taking into account the functional character of the scheme (1), we will consider the

[3] Compare with: "We can never be concerned only with the references of a sentence; but again the mere thought alone yields no knowledge, but only thought together with its reference, i.e. its true value" [Fre 892].

representation of knowledge functionally as follows:

$$\textsf{Representation: Information} \longrightarrow \textsf{Data Type.}$$

However, we cannot (and do not want to) avoid the notion of situation at all. We just move it from the outer world into the world of the computer's memory. As a matter of fact, when information enters the computer, it is in a certain state of the data type of knowledge. Therefore, the previous scheme turns out into the following one:

$$\textsf{Representation: Information} \times \textsf{Data Type} \longrightarrow \textsf{Data Type}$$

which articulates that a particular representation strategy is a set of two-sort computable partial functions taking into account both the structure of information flow and the structure of computer-represented data of knowledge.

Let us denote the data type of computer's knowledge via $\mathcal{D}$ with elements ε, ε', ... (probably with subscripts) and let $\Im$ be a set of all the truth values of propositions carrying information about the outer world. [4] Notice that the scheme (2) may be thought in the sense that facts just partially confirms propositions. That is to say, it is possible the case when the proposition A is comfirmed with two different truth values τ_1 and τ_2 according to the two groups of facts, i.e. both $A : \tau_1$ and $A : \tau_2$ hold. Thus, we might admit even the computer to tolerate contradictory information allowing $A : \boldsymbol{t}$ and $A : \boldsymbol{f}$. In general, we will admit the computable partial operation $[A : \tau]$ on $\mathcal{D}$ for any proposition A of a fixed formal language and $\tau \in \Im$. The operations of that kind we call *infons*, borrowing the name, but not the notion, from situation theory,[5] because the information content of $A : \tau$ is thought to be a connection of proposition A with unnamed facts that confirm it with the truth value τ.

To define the notion of infon more precisely, we have to understand what data type is. Recall that data types occur in programming languages, mostly as notions that originate from other mathematical notions such as, for example, as numbers. There are two main approaches to the notion of data type in computer science. First, *algebraic*, approach emphasises the treatment of

[4]We intentionally do not propose here any structure imposed on $\Im$, as it was done in [Mur 95a, Mur 95b].

[5]According to situation theory at least in its informal account, the notion of infon is to express items of information that characterize happening facts (cf. [Dev 91]).

data as being elements of a many-sort algebra (cf. [Wir 90]). Second, *axiomatic*, approach initiated by Dana Scott in [Sco 71] and undergone some changes since then aims to grasp the notion of approximation of data in changing environment. The last may be thought as the changing flow of information or that of state of the computer's knowledge. [6] In this paper we are interested in the latter aspect of the second approach. The approximation of knowledge data in the computer is thought, in turn, as giving more precise information about something rather than enlarging it.

Recall (cf. [DB 90, GS 90]) that a partially ordered set $\mathcal{P}$ is called *complete* if it has a bottom element ε_0 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 directed subset $D \subseteq \mathcal{P}$,

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

The bottom element is certainly compact.

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 $x \in \mathcal{P}$,

$$x = \sqcup\{y \,|y \in \mathcal{P},\, y \text{ is compact}\}. \tag{3}$$

A complete partially ordered set $\mathcal{P}$ is called a *Scott domain*, if equation (3) 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 (cf. [Mur 95b]).

We consider $\mathcal{D}_0$ as "real" elements of knowledge that can be represented in the computer, whereas $\mathcal{D}$ includes also "ideal" elements of knowledge that we need for adequate description of the outer world. As the equation (3) sets up, each ideal element can be approached with real elements. The exactness of the approach can be articulated in *Scott topology* on the domain $\mathcal{D}$. We refer the interested reader to [Sco 72, GHKLMS 80]. From now on, we assume $\mathcal{D}_0$ to be finitary objects, i.e. those which are capable to be used as the

[6] A brief comparison of the two approaches is contained in [LS 77].

computer's input. Also, recall that $\varepsilon_0 \in \mathcal{D}_0$. The interested reader can find in [Mur 95a, Mur 95b] examples of domains as being the knowledge carrier.

In accordance with the idea of approximation, we would want to limit ourselves with continuous function as potential knowledge transformers on a domain. We call a partial function $F : \mathcal{D} \to \mathcal{D}$ *Scott − continuous*, or simply *continuous, in* $x \in \mathcal{D}$ if $F(x)$ is defined, symbolically $F(x)\downarrow$, and for every directed set $\{x_i \,|\, i \in I\} \subseteq \mathcal{D}$,

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

provided that $\sqcup\{F(x_i) \,|\, i \in I\}$ exists.

One way to define $[A : \tau]$ is as follows. For every formula A and truth value $\tau \in \Im$, we pick an element ε^τ_A in $\mathcal{D}_0$. We conceive ε^τ_A as a minimal piece of information, according to which we can conclude, not depending on a current situation, that A has a truth value equal at least to τ and define[7]

$$[A : \tau](\varepsilon) \simeq \varepsilon \sqcup \varepsilon^\tau_A$$

for every $\varepsilon \in \mathcal{D}_0$ and then define

$$[A : \tau](x) \simeq \sqcup\{[A : \tau](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}$$

for every $x \in \mathcal{D}$, where $\simeq$ is Kleene's identity symbol for partially defined functions from [Kle 52]. Notice that $\{[A : \tau](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}$ is never empty, because $\varepsilon_0 \sqcup \varepsilon^\tau_A = \varepsilon^\tau_A$.

Theorem 1 *For every $x \in \mathcal{D}$, the following conditions are equivalent:*

$$\begin{aligned} i)&\quad [A : \tau](x)\downarrow;\\ ii)&\quad \{[A : \tau](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\} \textit{ has an upper bound;}\\ iii)&\quad \{[A : \tau](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\} \textit{ is directed.} \end{aligned}$$

Proof. The equivalence $(ii) \iff (iii)$ is an immediate consequence of the Lemma 3.20 in [DB 90] and a definition above.

The implication $(iii) \Rightarrow (i)$ is obvious.

Let us prove the implication $(i) \Rightarrow (iii)$. Assume

$$x_0 = \sqcup\{[A : \tau](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}.$$

[7]Compare this definition with the Theorem 4 in [Mur 94]; also see Section 7 in [KM 93].

Let ε_1, ε_2 be in $\mathcal{D}_0$ such that both $[A:\tau](\varepsilon_1)\downarrow$ and $[A:\tau](\varepsilon_2)\downarrow$ hold. Thus, x is an upper bound for $\{\varepsilon_1, \varepsilon_2\}$ and x_0 for $\{\varepsilon_1, \varepsilon_2, \varepsilon^{\tau}_{A}\}$, respectively. In virtue of the Lemma 3.22 in [DB 90], $\varepsilon_1 \sqcup \varepsilon_2$ exists and belongs to $\mathcal{D}_0$. It follows that x_0 is an upper bound for $\{\varepsilon_1 \sqcup \varepsilon_2, \varepsilon^{\tau}_{A}\}$. According to definitions, $\varepsilon_1 \sqcup \varepsilon_2 \leq x$ and $[A:\tau](\varepsilon_1 \sqcup \varepsilon_2) \simeq \varepsilon_1 \sqcup \varepsilon_2 \sqcup \varepsilon^{\tau}_{A}$. Consequently, $[A:\tau](\varepsilon_1 \sqcup \varepsilon_2)$ is defined. Thus, the set $\{[A:\tau](\varepsilon) \,|\varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}$ is directed.

Lemma 1 *If $[A:\tau](x)\downarrow$ and $x = \sqcup\{x_i \,|i \in I\}$ for some directed set $\{x_i \,|i \in I\}$, then $\sqcup\{[A:\tau](x_i) \,|i \in I\}$ exists.*

Proof. Indeed, we have the equation

$$[A:\tau](x) = \sqcup\{[A:\tau](\varepsilon) \,|\varepsilon \in \mathcal{D}_0, (\exists i \in I)(\varepsilon \leq x_i)\}. \tag{4}$$

Denote

$$x_0 = \sqcup\{[A:\tau](\varepsilon) \,|\varepsilon \in \mathcal{D}_0, (\exists i \in I)(\varepsilon \leq x_i)\}.$$

Thus, for every $i \in I$, if $\varepsilon \in \mathcal{D}_0$, $\varepsilon \leq x_i$ and $[A:\tau](\varepsilon)\downarrow$, then $[A:\tau](\varepsilon) \leq x_0$. Therefore, in view of the Lemma 3.20 in [DB 90], $\sqcup\{[A:\tau](x_i) \,|i \in I\}$ exists.

The following theorem establishes a necessary and sufficient condition, when $[A:\tau]$ is continuous in a point.

Theorem 2 *The function $[A:\tau]$ is continuous in x if and only if $[A:\tau]$ is defined in x and there is no directed set $\{x_i \,|i \in I\}$ such that $x = \sqcup\{x_i \,|i \in I\}$ and*

$$(\forall i \in I)([A:\tau](x_i) \text{ is not defined}). \tag{5}$$

Proof. In the case $x = \varepsilon_0$, the statement is obvious: both parts of the equivalence are true.

Let now $x \neq \varepsilon_0$. Suppose $[A:\tau]$ is continuous in x and, hence, defined in that point. Assume for some directed set $\{x_i \,|i \in I\}$, $x = \sqcup\{x_i \,|i \in I\}$ and the condition (5) is satisfied. Then

$$\sqcup\{[A:\tau](x_i) \,|i \in I\} = \sqcup\emptyset = \varepsilon_0.$$

We know that the set $\{[A:\tau](\varepsilon) \,|\varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}$ is non-empty. For instance, assume there is $\varepsilon' \in \mathcal{D}_0$ such that $\varepsilon' \leq x$ and $[A:\tau](\varepsilon')\downarrow$. If $\varepsilon^{\tau}_{A} = \varepsilon_0$, then

$[A:\tau](x) = x \neq \varepsilon_0$. If $\varepsilon^{\tau}_A \neq \varepsilon_0$, then $\varepsilon_0 < [A:\tau](\varepsilon') \leq [A:\tau](x)$. That is, $[A:\tau]$ is not continuous in x, that contradicts to the premise.

Suppose $[A:\tau](x)$ is defined, though $[A:\tau]$ is not continuous in x. That is to say, there is a directed set $\{x_i \,|i \in I\}$ such that, though $x = \sqcup\{x_i \,|i \in I\}$ and in virtue of the Lemma 1, $\sqcup\{[A:\tau](x_i) \,|i \in I\}$ exists,

$$[A:\tau](x) \neq \sqcup\{[A:\tau](x_i) \,|i \in I\}. \tag{6}$$

Furthermore, suppose that the following correlation holds:

$$(\forall \varepsilon \in \mathcal{D}_0)(\varepsilon \leq x \Rightarrow (\exists i \in I)(\varepsilon \leq x_i \& [A:\tau](x_i)\downarrow). \tag{7}$$

Then

$$[A:\tau](x) \leq \sqcup\{[A:\tau](x_i) \,|i \in I\}.$$

The converse inequality is obvious. Thus, we receive a contradiction to (6). Consequently, (7) is not true. It follows that there is $\varepsilon_1 \in \mathcal{D}_0$ such that $\varepsilon_1 \leq x$ and

$$(\forall i \in I)(\varepsilon_1 \leq x_i \Rightarrow \text{ not } [A:\tau](x_i)\downarrow).$$

Denote $J = \{i \,|i \in I, \varepsilon_1 \leq x_i\}$. Notice that the set $\{x_j \,|j \in J\}$ is non-empty and directed. Indeed, the former follows from the correlation

$$\varepsilon_1 \leq x = \sqcup\{x_i \,|i \in I\},$$

that ε_1 is compact and the premise of that $\{x_i \,|i \in I\}$ is directed. That premise also implies the latter. Thus, $\sqcup\{x_j \,|j \in J\}$ exists and is less or equal to x. On the other hand, for every $i \in I$ there is $j \in J$ such that $x_i \leq x_j$. Consequently,

$$x \leq \sqcup\{x_i \,|i \in I\} \leq \sqcup\{x_j \,|j \in J\}.$$

Going over to the definition of another kind of knowledge transformers, *constraints*, we recall that in situation theory, where we borrow the term, they are to regulate relationship between information types [Bar 92]. On our part, we consider first the conditions of sort $A:\tau_a \rightarrow B:\tau_b$ for any formulas A and B and $\tau_a, \tau_b \in \Im$, that a knowledge maintenance builder may want to impose, proposing that the system moves to a state satisfying $B:\tau_b$ as soon as a current state satisfies $A:\tau_a$. We also suppose that move to be performed in a minimal way, that is to say, with minimal "distortion". [8] Accordingly,

[8] That idea is due to N. Belnap (cf. [Bel 75]).

we link with a condition $A:\tau_a \to B:\tau_b$ a partial operation $[A:\tau_a \to B:\tau_b]$ on $\mathcal{D}_0$, being then extended to the entire $\mathcal{D}$, getting a proper constraint.

To satisfy the idea of minimal way, we suppose to be able for every formula A, value $\tau \in \Im$ and element $\varepsilon \in \mathcal{D}_0$, to have two "refinements" of the last in $\mathcal{D}_0$ — $d_A^{\tau+}(\varepsilon)$ and $d_A^{\tau-}(\varepsilon)$. The former carries all information contained in ε about $A:\tau$ and the latter carries the complementation to that information so that both "exhaust" ε. Let us go to precise definitions.

If a domain $\mathcal{D}$ does not have already a greatest element, add it denoted by $\mathbf{1}$ to $\mathcal{D}$ for the sake of convenience. Fix two (computable) functions, denoted $d_A^{\tau_a+}$ and $d_A^{\tau_a-}$ for any formula A and $\tau_a \in \Im$, from $\mathcal{D}_0$ into $\mathcal{D}_0 \cup \{\mathbf{1}\}$, satisfying the following conditions:

- $d_A^{\tau_a+}(\varepsilon_0) \in \{\varepsilon_0, \mathbf{1}\}$;
- $\varepsilon = d_A^{\tau_a+}(\varepsilon) \sqcap d_A^{\tau_a-}(\varepsilon)$;
- $d_A^{\tau_a+}\big(\varepsilon^{\tau}_A\big) = \varepsilon^{\tau}_A$ and $d_A^{\tau_a-}\big(\varepsilon^{\tau}_A\big) = \mathbf{1}$;
- $d_A^{\tau_a+}(\varepsilon) = \varepsilon \iff d_A^{\tau_a-}(\varepsilon) = \mathbf{1}$;
- $d_A^{\tau_a-}(\varepsilon) = \varepsilon \iff d_A^{\tau_a+}(\varepsilon) = \mathbf{1}$;
- $d_A^{\tau_a+}(\varepsilon_1 \sqcup \varepsilon_2) \leq d_A^{\tau_a+}(\varepsilon_1) \sqcup d_A^{\tau_a+}(\varepsilon_2)$, providing $\sqcup$-sums exist.

Define[9] for every $\varepsilon \in \mathcal{D}_0$,

$$[A:\tau_a \to B:\tau_b](\varepsilon) \simeq d_A^{\tau_a-}(\varepsilon) \sqcap [B:\tau_b]\Big(d_A^{\tau_a+}(\varepsilon)\Big).$$

Notice that for every $\varepsilon \in \mathcal{D}_0$, $[A:\tau_a \to B:\tau_b](\varepsilon)$ never receives the value $\mathbf{1}$. Also, note:

$$[A:\tau_a \to B:\tau_b](\varepsilon_0) = \begin{cases} \varepsilon_0 & \text{if } d_A^{\tau_a-}(\varepsilon_0) = \varepsilon_0 \\ \varepsilon_B^{\tau_b} & \text{if } d_A^{\tau_a-}(\varepsilon_0) = \mathbf{1}. \end{cases}$$

It implies that the set $\{[A:\tau_a \to B:\tau_b](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}$ is non-empty for every $x \in \mathcal{D}$.

[9]Compare with the proof of the Theorem 4 in [Mur 94].

Furthermore, define for every $x \in \mathcal{D}$,

$$[A : \tau_a \to B : \tau_b](x) \simeq \sqcup\{[A : \tau_a \to B : \tau_b](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}.$$

Write out the following property:

$$[B : \tau_b](\varepsilon_1) \downarrow \& [B : \tau_b](\varepsilon_2) \downarrow \Rightarrow \varepsilon_1 \sqcup \varepsilon_2 \sqcup \varepsilon_B^{\tau_b} \text{ exists} \tag{8}$$

for every $\varepsilon_1, \varepsilon_2 \in \mathcal{D}_0$.

Theorem 3 *Providing the property (8), for every $x \in \mathcal{D}$, the following conditions are equivalent:*

i) $[A : \tau_a \to B : \tau_b](x) \downarrow$;
ii) $\{[A : \tau_a \to B : \tau_b](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}$ *has an upper bound;*
iii) $\{[A : \tau_a \to B : \tau_b](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}$ *is directed.*

Proof. All implications, except $(i) \Rightarrow (iii)$, are proved in the same manner as corresponding implications in the Theorem 1 above. Prove the implication $(i) \Rightarrow (iii)$.

First of all, recall that the set $\{[A : \tau_a \to B : \tau_b](\varepsilon) \,|\, \varepsilon \in \mathcal{D}_0, \varepsilon \leq x\}$ is non-empty.

Furthermore, assume for $\varepsilon_1, \varepsilon_2 \in \mathcal{D}_0$ and such that $\varepsilon_1, \varepsilon_2 \leq x$, both $[A : \tau_a \to B : \tau_b](\varepsilon_1) \downarrow$ and $[A : \tau_a \to B : \tau_b](\varepsilon_2) \downarrow$ hold. Then, first, in virtue of the Lemma 3.22 in [DB 90], $\varepsilon_1 \sqcup \varepsilon_2$ exists, belongs to $\mathcal{D}_0$ and less or equal to x, and, second, both $[B : \tau_b]\left(d_A^{\tau_a+}(\varepsilon_1)\right) \downarrow$ and $[B : \tau_b]\left(d_A^{\tau_a+}(\varepsilon_2)\right) \downarrow$ hold, either. According to (8), we receive that

$$d_A^{\tau_a+}(\varepsilon_1) \sqcup d_A^{\tau_a+}(\varepsilon_2) \sqcup \varepsilon_B^{\tau_b}$$

exists. Consequently, $d_A^{\tau_a+}(\varepsilon_1) \sqcup d_A^{\tau_a+}(\varepsilon_2) \sqcup \varepsilon_B^{\tau_b}$ is an upper bound for $\{d_A^{\tau_a+}(\varepsilon_1 \sqcup \varepsilon_2), \varepsilon_B^{\tau_b}\}$. Therefore, $[A : \tau_a \to B : \tau_b](\varepsilon_1 \sqcup \varepsilon_2)$ is defined.

Write out the following property:

$$(\forall \varepsilon \in \mathcal{D}_0)(\varepsilon \leq x \& [A : \tau_a \to B : \tau_b](\varepsilon) \downarrow \Rightarrow [A : \tau_a \to B : \tau_b](\varepsilon) = \varepsilon_0) \Rightarrow x = \varepsilon_0. \tag{9}$$

Theorem 4 *Let (9) hold for x. Then $[A:\tau_a \rightarrow B:\tau_b]$ is continuous in x if and only if $[A:\tau_a \rightarrow B:\tau_b](x)$ is defined and there is no directed set $\{x_i \,|i \in I\}$ such that $x = \sqcup\{x_i \,|i \in I\}$ and*

$$(\forall i \in I)([A:\tau_a \rightarrow B:\tau_b](x_i) \textit{ is not defined}). \tag{10}$$

Proof. Let $x = \varepsilon_0$. Then both parts of the equivalence in the theorem's statement are true.

Assume $x \neq \varepsilon_0$. Let $[A:\tau_a \rightarrow B:\tau_b]$ be continuous in x, the set $\{x_i \,|i \in I\}$ be directed, satisfying (10), and $x = \sqcup\{x_i \,|i \in I\}$. Then

$$\sqcup\{[A:\tau_a \rightarrow B:\tau_b](x_i) \,|i \in I\} = \sqcup\emptyset = \varepsilon_0.$$

It implies that the premise of (9) is true. Consequently, $x = \varepsilon_0$. A contradiction.

The proof of the "if" part is the same as that in the Theorem 2.

We aim to consider the issues of computability of infons and constraints elsewhere. It is clear as of this moment that we have to go over to a more specific structure of domain. And if we want to define the domain structure within the mode of many-valued logic, we have to specify a structure of $\mathfrak{I}$ in some way as well, for example, as it was done in [Mur 95a, Mur 95b]. Also, we need to find a "justification" for the use of conditions (8) and (9), respectively, in the Theorems 3 and 4 above.

References

[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 .

[BP 83] J.Barwise and J.Perry, Situations and Attitudes, The MIT Press, 1983.

[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].

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

[Dev 91] K.Devlin, Logic and Information, Cambridge University Press, Cambridge, 1991.

[Fre 892] G.Frege, Über Sinn und Bedeutung, 1892. Trans.: On Sense and Reference, in: P. Geach and M. Black (eds.), *Translations from the Philosophical Writings of Gottlob Frege*, Basil Blackwell, Oxford, 1952.

[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.

[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.

[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.

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

[LS 77] D.J.Lehmann and M.B.Smyth, Data Types, in: *Proc. 18th Conf. on Foundations of Computer Science*, Providence, RI, 1977, pp. 7–12.

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

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

[Mur 95b] A.Yu.Muravitsky, Foundations for Knowledge Maintenance Systems, I, 1995, to be published.

[Rus 05] B.Russell, On Denoting, 1905. Reprinted in: R.C.Marsh (ed.), *Logic and Knowledge*, George Allen and Unwin, London, 1956, pp. 41–56.

[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, pp. 175–281.

[Sco 71] D.S.Scott, Outline of a Mathematical Theory of Computation, in: *Proceedings of Princeton Conference on Information Science*, 1971, pp. 169–176.

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

[Wir 90] M.Wirsing, Algebraic Specification, in: J. van Leeuwen (ed.) *Handbook of Theoretical Computer Science*, Volume B: "Formal Models and Semantics", 1990, Elsevier, pp. 675–788.