REASONING WITH MULTILEVEL CONTEXTS IN SEMANTIC METANETWORK

Vagan Y. Terziyan (*), Seppo Puuronen (**)

(*) MetaIntelligence Lab., State Technical University of Radioelectronics
14 Lenina av., 310166, Kharkov, Ukraine
phone: +380 572 409 890; fax: +380 572 409 113;
e-mail: vagan@kture.cit-ua.net; vagan@jytko.jyu.fi

(**) Department of Computer Science and Information Systems,
University of Jyväskylä, P.O.Box 35, SF-40351, Jyväskylä, Finland
phone: +358 14 603 028; fax +358 14 603 011; e-mail: sepi@jytko.jyu.fi

Abstract

In this paper, a multilevel semantic network is proposed to be used to represent knowledge within several levels of contexts. The zero level of representation is semantic network that includes knowledge about basic domain objects and their relations. The first level of presentation uses semantic network to represent contexts and their relationships. The second level presents relationships of metacontexts i.e. contexts of contexts, and so on at the higher levels. The topmost level includes knowledge which is considered to be “truth” in all the contexts. Thus a semantic metanetwork is the hierarchical set of semantic networks above each other so that relations of each previous level are context objects of the next level. Such representation allows to reason with contexts towards solution of the following problems: to derive knowledge interpreted using all known levels of its context; to derive unknown knowledge when interpretation of it in some context and the context itself are known; to derive unknown knowledge about a context when it is known how the knowledge is interpreted in this context; to transform knowledge from one context to another. Possible transformations with contexts are described using special algebra. Equations of the algebra are discussed and used to reason with this multilevel context structure.

Keywords: multilevel context, metasemantic network, formalization, decontextualization, context recognition, lifting

1. Introduction

It is generally accepted that knowledge has a contextual component. Acquisition, representation, and exploitation of knowledge in context would have a major contribution in knowledge representation, knowledge acquisition, and explanation, as Brezillon and Abu-Hakima supposed in [5]. Among the advantages of the use of contexts in knowledge representation and reasoning Akman and Surav [1] mentioned the following: economy of representation, more competent reasoning, allowance for inconsistent knowledge bases, resolving of lexical ambiguity and flexible entailment. Brezillon and Cases noticed however in [6] that knowledge-based systems do not use correctly their knowledge. Knowledge being acquired from human experts does not usually include its context.

Contextual component of knowledge is closely connected with eliciting expertise from one or more experts in order to construct a single knowledge base (or, for example as in [4], for cooperative building of explanations). Could the overlapping knowledge, obtained from multiple sources, be described in such a way that it becomes context or even process independent? In [25], Taylor at al. gives the negative answer. Certainly there have been inference engines produced that were subsequently applied to related domains, but in general the sets of rules have been different. If more than one expert are available, one must either select the opinion of the best expert or pool the experts’ judgements. It is assumed that when experts’ judgements are pooled, collectively they offer sufficient cues leading to the building of a comprehensive theory.

Very important questions have been raised in [5]. Does context simplify or complicate the construction of a knowledge base? Is context an object of the domain? Can we move from one context to the next one [17]? What are the possible formalisms that seem to allow explicit representation of context?

McCarthy [17] illustrates how a reasoning system can utilize contexts to incorporate information from a general common-sense knowledge base into other specialized knowledge bases. The basic relation used is *ist (c,p)*, which asserts that the proposition *p* is true in the context *c*. He introduces contexts as abstract mathematical entities with properties and relations. One remark is that it would be useful to have a formal theory of the natural phenomenon of context using logic for representation. Later in [16] he noticed that statements about contexts are themselves in contexts. Contexts may be treated as mathematical structures of different properties and also may have different relations with other contexts. The formulas express relations among contexts would be primary rather than propositions true in the contexts. According to McCarthy there is no general context in which all the stated axioms always hold.

Different uses of contexts were analyzed by Guha in [14]. In his approach all axioms and statements are not universally true, they are only true in contexts. Such contexts he calls microtheories which make different assumptions about the world. He has developed some very general lifting rules using which different microtheories can be integrated. He also uses contexts for integrating multiple databases and handling mutual inconsistencies between databases.

Buvac at al. investigated the simple logical in [7, 8] and semantic (in [9]) properties of contexts. This formal theory of context and use of quantificational logic enables presentation of relations between contexts, operations on contexts, and state lifting rules of facts in different contexts. The quantificational logic of context [10], for example, enables to state that the formula ψ is true in all contexts which satisfy some property $p(x)$ as follows: $(\forall v) p(v) \rightarrow ist(v, \psi)$. Each context is considered to have a set of propositional atoms as its own vocabulary. It was shown that the acceptance of outermost context simplifies the metamathematics of the contexts.

Attardi and Simi [2] offer a viewpoint representation of context related formalisms. Viewpoints denote sets of statements which represent the assumptions of a theory. The basic relation in their formalization: $in('A', vp)$, where *vp* is a viewpoint expression, can be interpreted as “statement *A* is entailed by the assumption denoted by *vp*”. Operations between viewpoints are carried out with metalevel rules. Relation $Holds(A,s) = in(A, vp(s))$, which defines the set of facts *vp(s)* holding in a situation *s*, is used to handle situations as sets of basic facts.

Edmonds [11] describe a simple extension of semantic nets. The nodes are labeled with directed arcs and the directed arcs can lead to other arcs as well as nodes. In this model, contexts are not differentiated as special objects, but rather that some nodes to a greater or lesser extent have roles as encoders of contextual information. This formulation is shown to be expressive enough to capture several aspects of contexts including reasoning and generalization in contexts. It is not claimed that this is a model of any type of context found in human activity.

After making their comparison of approaches towards formalizing context, Akman and Surav [1] concluded that the idea of formalizing context seems to have caught on and produce good theoretical outcomes and the area of innovative applications remains relatively unexplored.

As one can imagine looking context related references that the context itself has so many different meanings relatively to the goals of its study, application areas, formal methods used and so on. To understand and apply the research results in context area one has to be involved to the context in which these results have been obtained. The goal of this paper is to make view to a world of contexts, their properties, relationships and rules as to a new domain which also needs a context for interpretation. Context of contexts (metacontext) can be considered recursively and has several levels of context worlds, one above another, limited in the top by a *universal context* world. This last world supposed to be such that knowledge about it always remains the same in every known context. Representation of such multilevel world needs special multilevel semantic structure. With such representation we use term *semantic metanetwork*.

The idea of a semantic metanetwork is generally acknowledged by Puuronen and Terziyan in [18] and further developed in [3, 26]. In [18] objects of a higher level representation were treated as “birth” and “death” rules for relations between objects of the lower level. Thus the semantic metanetwork together with the description of the basic domain contains the rules of its change in time. Sometimes it is also necessary to change the rules. This is done by metarules of the next level of a semantic metanetwork. The more complex the dynamics of the described domain is, the more metalevels a semantic metanetwork contains.

This paper discusses another use of a semantic metanetwork. The zero level of representation is a semantic network that includes knowledge about basic domain objects and their relations in various contexts. The first level of representation uses a semantic network to present knowledge about contexts and their relationships. Relationships among contexts are considered in some other contexts - metacontexts. The second level of representation defines relationships between metacontexts, and the same for each next level. The topmost level includes knowledge which is considered to be “truth” in every context.

What are the main uses of semantic metanetworks? The need of metaknowledge is broadly acknowledged in knowledge engineering and its main use is to help inference (metareasoning has been dealt for example in [22]). Shastri presents in his article [23] two important kinds of inference: inheritance and recognition. Woods [27] discusses very deeply the classification of a concept with respect to a given taxonomic structure. Shapiro [24] discussed also inference as reduction inference and path-based inference. A context can be considered as a mechanism for reasoning, too. One advantage of making context explicit in a representation is the capability to inference within and across contexts, and thus explicitly make the changes in reasoning across contexts [17].

Our goal is to present methods of reasoning using knowledge in its context formally represented by semantic metanetworks. These problems are: how to derive knowledge about any relation which is interpreted in the highest level of context; how to derive the initial knowledge about any relation if it is known the context in which this relation has been interpreted and the result of interpretation; how to derive knowledge about any unknown context by analyzing its effect to the initial knowledge; how to transform knowledge from one context to another.

2. A Semantic Metanetwork

In this chapter we give formal definition of a semantic metanetwork and show one example.

Semantic metanetwork is a formal representation of knowledge with contexts.

Knowledge of basic domain is represented by a semantic network with nodes (domain objects) and links (relations between domain objects). There are two types of relations: relations connecting two domain objects (arrow between nodes) and relations of domain objects with themselves (arrow from a node to itself). The second type of relation presents a property of an object.

We also consider a context as a domain object. Sometimes it is necessary to describe knowledge about contexts, too. Knowledge about properties and relations of contexts compose a new semantic network which is considered to describe the next level of knowledge representation. Levels of such representation have relationship with each other because of correspondence: relation of each previous level - the context (metacontext) of the next level in which this relation occurs.

Formally metanetwork is a quadruple <*A*,*L*,*S*,*D*>, where *A* is a set of objects which consists of the subset of basic domain objects $A_i^0, i = 1,\ldots,n_1$, and several subsets of contexts $A_i^{(d)}, i = 1,\ldots,n_d$, and $d = 1,\ldots,klev$ identifies the level of the metanetwork where the context appears; *L* is a set of unique names of relations $L_k^{(d)}, k = 1,\ldots,m_d$ and $d = 0,\ldots,klev$ identifies the level of the metanetwork where the relation appears; *S* is the set of relations $S_r^{(d)} = P(A_i^{(d)}, L_k^{(d)}, A_j^{(d)}), r = 1,\ldots,l_d$ composed in each level so that: $S^{(d)} = \bigwedge_r S_r^{(d)}$, and $P(A_i^{(d)}, L_k^{(d)}, A_j^{(d)})$ is true when there is the relation $L_k^{(d)} \in L$ between the objects $A_i^{(d)} and\, A_j^{(d)}, (A_i^{(d)}, A_j^{(d)} \in A)$ at the level *d*; and *D* is the set of context predicates

$D_r^{(d)} = ist(A_i^{(d+1)}, S_r^{(d)})$ connecting contexts of the level *d+1* to the relations of the level *d* and $ist(A_i^{(d+1)}, S_r^{(d)})$ is true if the relation $S_r^{(d)}$ holds in the context $A_i^{(d+1)}$.

Let us consider an example of a semantic metanetwork presented in Figure 1.

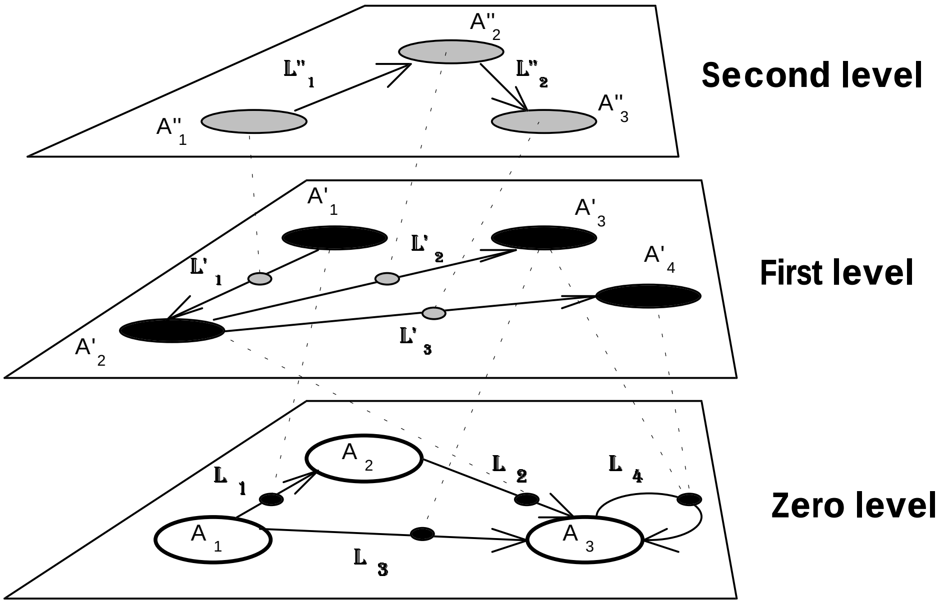

Fig. 1. An example of a semantic metanetwork

The metanetwork in Figure 1 can be described by the following expressions:

$$A = \{A^0, A', A''\}, A^0 = \{A_1, A_2, A_3\}, A' = \{A_1', A_2', A_3'\}, A'' = \{A_1'', A_2'', A_3''\}$$

are the objects' set and its three subsets accordingly to levels;

$$L = \{L^0, L', L''\}, L^0 = \{L_1, L_2, L_3, L_4\}, L' = \{L_1', L_2', L_3'\}, L'' = \{L_1'', L_2''\}$$

are the set of relations' names and its three subsets accordingly to levels;

$$S = \{S^0, S', S''\},\ S^0 = \{S_1, S_2, S_3, S_4\}, S' = \{S_1', S_2', S_3'\}, S'' = \{S_1'', S_2''\}$$

are the set of relations and its three subsets accordingly to levels;

$$S_1 = P(A_1, L_1, A_2), S_2 = P(A_2, L_2, A_3), S_3 = P(A_1, L_3, A_3), S_4 = P(A_3, L_4, A_3);$$

$$S_1' = P(A_1', L_1', A_2'), S_2' = P(A_2', L_2', A_3'), S_3' = P(A_2', L_3', A_4');\ S_1'' = P(A_1'', L_1'', A_2''), S_2'' = P(A_2'', L_2'', A_3'')$$

are the relations of all the three levels in a predicate form;

$$D = \{D^0, D'\},\ D^0 = \{D_1, D_2, D_3, D_4, D_5\}, D' = \{D_1', D_2', D_3'\}$$

are the set of context predicates and its two subsets accordingly to levels;

$$D_1 = ist(A_1', S_1), D_2 = ist(A_2', S_2), D_3 = ist(A_3', S_3), D_4 = ist(A_3', S_4), D_5 = ist(A_4', S_4)$$

are the context predicates describing context relationships between zero and first levels;

$$D_1' = ist(A_1'', S_1'), D_2' = ist(A_2'', S_2'), D_3' = ist(A_3'', S_3')$$

are the context predicates describing context relationships between first and second levels.

In Figure 1 and in the formal description of the example, the objects A_1, A_2, A_3 - are the basic domain objects or objects of the basic level of the semantic metanetwork. The basic domain relations are: S_1 between objects A_1 and A_2 named by L_1; S_2 between objects A_2 and A_3 named by L_2; S_3

between objects A_1 and A_3 named by L_3; S_4 between objects A_3 and itself named by L_3, or another words L_3 is the name of property of the object A_3. The relation S_1 exists in the context named $A_1^{'}$, and also relations S_2 - S_3 exist in the contexts named $A_2^{'} - A_3^{'}$ respectively. The relation S_4 exists if the two contexts $A_3^{'}, A_4^{'}$ are taking place together. The contexts $A_1^{'} - A_4^{'}$ are the first level contexts and they belong to the first level of the semantic metanetwork. In the example, contexts also have their relationships $S_1^{'} - S_3^{'}$ named by $L_1^{'} - L_3^{'}$ respectively. The relation $S_1^{'}$ exists in the metacontext named $A_1^{''}$, and also relations $S_2^{'}, S_3^{'}$ exist in the metacontexts named $A_2^{''}, A_3^{''}$ respectively. The metacontexts $A_1^{''} - A_3^{''}$ are the second level contexts and they belong to the second level of the semantic metanetwork. Metacontexts also have their relationships $S_1^{''}, S_2^{''}$ named by $L_1^{''}, L_2^{''}$ respectively. In the example, there is not any context (metametacontext) to interpret relations $S_1^{''}, S_2^{''}$ which are considered to be valid in any known context. Thus the example describes the semantic metanetwork which is obtained by combination of three semantic networks.

3. An Algebra Within One Level of a Semantic Metanetwork

We will consider an Algebra defined on the set L of names of semantic relations. In this chapter we focus on operation and equations within one level of a semantic metanetwork. There are four basic operations upon L-set (semantic operations).

3.1. Semantic constants and operations

We will define basic semantic operations of the Algebra by giving formal definition and properties and also by giving graphical interpretation. Further in formulas we will use "$\Leftrightarrow$" to mark the logical equivalence between any two predicates, and "$\Rightarrow$" to mark the logical consequence between two predicates; "$\equiv$" between names of relations or objects to mark equality of semantic meanings of these two names, "$\neq$" between names of relations to mark inequality of semantic meanings of these two names. Proof for some of properties is also presented. It uses, together with formal definitions of operations, the following notation:

$$(P(A_i, L_k, A_j) \Leftrightarrow P(A_i, L_m, A_j)) \Leftrightarrow L_k \equiv L_m .$$

3.1.1. Semantic constants

It is supposed that two objects represented in semantic network by two different circles with two different names A_i and A_j are different objects. Even if there is not any knowledge about relationship between these objects, then at least it is still true that we know the fact of the difference between the objects. We consider knowledge about the difference as semantic constant *DIF* which denotes the relation between every pair of different objects:

$$\forall A_i, A_{j(j \neq i)} (P(A_i, DIF, A_j) \Leftrightarrow true) .$$

On the other hand the above formula can be interpreted that there is no difference between every object and itself. This statement defines the another semantic constant of equivalence *SAME*:

$$\forall A_i (P(A_i, SAME, A_i) \Leftrightarrow true) .$$

The last formula also means that if there is not any knowledge about properties of some object, than at least one property always holds - to be same to itself.

We define *NIL* relation as relation which never holds between any two objects or among properties of any object by the following way:

$$\forall A_i, A_{j(j\neq i)} (\neg P(A_i, DIF, A_j) \Leftrightarrow P(A_i, NIL, A_j) \Leftrightarrow false),$$
$$\forall A_i (\neg P(A_i, SAME, A_i) \Leftrightarrow P(A_i, NIL, A_i) \Leftrightarrow false).$$

When the semantic network is being built the following assumption is used. If there exist two different portions of knowledge and there is not explicitly said that they belong to the description of the same object then it is assumed that they describe different objects. Sometimes it happens that semantics (properties and relationships) of these two objects are always the same. When it happened then the two objects should be connected by the binary equivalent of *SAME* relation using the following definition:

$$(P(A_s, L_k, A_i) \Leftrightarrow P(A_s, L_k, A_j), \forall A_s \in A, \forall L_k \in L) \Leftrightarrow P(A_i, SAME^b, A_j),$$

where $SAME^b$ means the binary equivalent of *SAME* relation. This relation does not mean the negation of *DIF* relation, because it means only that two different objects have the same semantics.

3.1.2. Semantic inversion

Formally semantic inversion can be defined by the following equation:

$$P(A_i, L_k, A_j) \Leftrightarrow P(A_j, \tilde{L}_k, A_i),$$

where $\tilde{L}_k$ is the new inverse relation. It will be named with unique symbol L_m and added to the set *L*. For example, if L_k = <*to_punish*>, then L_m = <*to_be_punished*> and $L_m \equiv \tilde{L}_k$. Similarly, let L_n = <*to_be_on_the_left_side_of*>, then L_q = <*to_be_on_the_right_side_of*> and $L_q \equiv \tilde{L}_n$.

If the relation in the definition means a property of an object then its inversion is equal to the original one: $P(A_i, L_k, A_i) \Leftrightarrow P(A_i, \tilde{L}_k, A_i)$ which means for unary relation: $L_k \equiv \tilde{L}_k$.

The obvious property for the semantic inversion operation is *double inversion*: $\tilde{\tilde{L}}_k \equiv L_k$.

3.1.3. Semantic negation

The semantic negation operation means changing the name of a relation if the value of appropriate predicate is false. This is defined in a following way:

$$\neg P(A_i, L_k, A_j) \Leftrightarrow P(A_i, \bar{L}_k, A_j),$$

where $\bar{L}_k$ is the new relation. It can be named with unique symbol L_m and added to the set *L*. If, for example, *P(<Mary>, <to_love>, <Tom>) = false*, then it means the same as: *P(<Mary>, <not_to_love>, <Tom>) = true*. Thus, if L_k = <*to_love*> and L_m = <*not_to_love*>, then $L_m \equiv \bar{L}_k$.

Properties:

- double negation: $\bar{\bar{L}}_k \equiv L_k$;
- negation of inversion: $\bar{\tilde{L}}_k \equiv \tilde{\bar{L}}_k$; Proof:

$$(P(A_i, \bar{\tilde{L}}_k, A_j) \Leftrightarrow \neg P(A_i, \tilde{L}_k, A_j) \Leftrightarrow \neg P(A_j, L_k, A_i) \Leftrightarrow P(A_j, \bar{L}_k, A_i) \Leftrightarrow P(A_i, \tilde{\bar{L}}_k, A_j)) \Leftrightarrow \bar{\tilde{L}}_k \equiv \tilde{\bar{L}}_k.$$

3.1.4. Semantic multiplication

The semantic multiplication operation defines the name of unknown relation between a pair of objects A_i and A_j if there exists the third object A_s which is connected with both of these two objects. The formal definition is as follows:

$$P(A_i, L_k, A_s) \wedge P(A_s, L_n, A_j) \Leftrightarrow P(A_i, L_k * L_n, A_j),$$

where $L_k * L_n$ is the new relation. It can be named with unique symbol L_m and added to the set *L*. If, for example, it is true that *P(<Mary>, <to_be_married_with>, <Tom>)* and *P(<Tom>,*

<to_have_mother>, *<Diana>)*, then it is also true that: *P(<Mary>*, *<to_have_mother-in-law>*, *<Diana>).* Thus, if L_k = *<to_be_married_with>*, L_n = *<to_have_mother>, and* L_m = *<to_have_mother-in-law>*, then $L_m \equiv L_k * L_n$.

Properties:

- non-commutativity $\neg(L_k * L_n \equiv L_n * L_k), \forall L_k, L_n (L_k \neq L_n \neq DIF \neq SAME^b)$.
- transitivity $L_k * (L_n * L_m) \equiv (L_k * L_n) * L_m$; Proof:

$$(P(A_i, L_k * (L_n * L_m), A_j) \Leftrightarrow P(A_i, L_k, A_s) \wedge P(A_s, L_n * L_m, A_j) \Leftrightarrow P(A_i, L_k, A_s) \wedge$$
$$\wedge P(A_s, L_n, A_t) \wedge P(A_t, L_m, A_j) \Leftrightarrow P(A_i, L_k * L_n, A_t) \wedge P(A_t, L_m, A_j) \Leftrightarrow$$
$$\Leftrightarrow P(A_i, (L_k * L_n) * L_m, A_j)) \Leftrightarrow L_k * (L_n * L_m) \equiv (L_k * L_n) * L_m;$$

- inversion over multiplication ${}^{\sim}(L_k * L_n) \equiv \tilde{L}_n * \tilde{L}_k$; Proof:

$$(P(A_j, L_k * L_n, A_i) \Leftrightarrow P(A_j, L_k, A_s) \wedge P(A_s, L_n, A_i) \Leftrightarrow P(A_s, \tilde{L}_k, A_j) \wedge P(A_i, \tilde{L}_n, A_s) \Leftrightarrow$$
$$\Leftrightarrow P(A_i, \tilde{L}_n, A_s) \wedge P(A_s, \tilde{L}_k, A_j) \Leftrightarrow P(A_i, \tilde{L}_n * \tilde{L}_k, A_j)) \Leftrightarrow {}^{\sim}(L_k * L_n) \equiv \tilde{L}_n * \tilde{L}_k.$$

3.1.5. Semantic addition

The semantic addition operation defines the name of relation between a pair of objects A_i and A_j as a combination of two relations between these two objects. Formally:

$$P(A_i, L_k, A_j) \wedge P(A_i, L_n, A_j) \Leftrightarrow P(A_i, L_k + L_n, A_j),$$

where $L_k + L_n$ is the new relation. It can be named with unique symbol L_m and added to the set L. If, for example, it is true that *P(<Mary>, <to_give_birth_to>, <Tom>)* and *P(<Mary>*, *<to_take_care_of>*, *<Tom>)*, then: *P(<Mary>, <to_be_mother_of>, <Tom>)* is also true. Thus, if L_k = *<to_give_birth_to>*, L_n = *<to_take_care_of>, and* $L_m = L_k + L_n$ = *<to_be_mother_of>.*

It is also possible to sum properties. If, for example, it is true that *P(<Tom>, <to_be_clever>*, *<Tom>)* and *P(<Tom>, <to_be_rich>,<Tom>),* then it is also true that: *P(<Tom>, <to_be_clever_and_rich>, <Tom>).*

Properties:

- commutativity $L_k + L_n \equiv L_n + L_k$;
- transitivity $L_k + (L_n + L_m) \equiv (L_k + L_n) + L_m$;
- reflexivity $L_k + L_k \equiv L_k$;
- inversion over sum ${}^{\sim}(L_k + L_n) \equiv \tilde{L}_k + \tilde{L}_n$;
- distributivity (left) $L_k * (L_m + L_n) \equiv L_k * L_m + L_k * L_n$; Proof:

$$(P(A_i, L_k * (L_m + L_n), A_j) \Leftrightarrow P(A_i, L_k, A_s) \wedge P(A_s, L_m + L_n, A_j) \Leftrightarrow$$
$$\Leftrightarrow P(A_i, L_k, A_s) \wedge P(A_s, L_m, A_j) \wedge P(A_s, L_n, A_j) \Leftrightarrow P(A_i, L_k, A_s) \wedge P(A_i, L_k, A_s) \wedge$$
$$\wedge P(A_s, L_m, A_j) \wedge P(A_s, L_n, A_j) \Leftrightarrow (P(A_i, L_k, A_s) \wedge P(A_s, L_m, A_j)) \wedge$$
$$\wedge (P(A_i, L_k, A_s) \wedge P(A_s, L_n, A_j)) \Leftrightarrow P(A_i, L_k * L_m, A_j) \wedge P(A_i, L_k * L_n, A_j) \Leftrightarrow$$
$$\Leftrightarrow P(A_i, L_k * L_m + L_k * L_n, A_j)) \Leftrightarrow L_k * (L_m + L_n) \equiv L_k * L_m + L_k * L_n;$$

- distributivity (right) $(L_k + L_m) * L_n \equiv L_k * L_n + L_m * L_n$.

3.1.6. Semantic closeness

We define the L-set $L: \{L_1, L_2, \ldots L_n\}$ as semantically closed set of relations' names if the result of any semantic operation with any operands from the L-set also belongs to the L-set. Formally we define semantic closeness as follows:

$$\tilde{L}_k \in L, \forall L_k \in L; \quad \overline{L}_k \in L, \forall L_k \in L; \quad L_k * L_m \in L, \forall L_k, L_m \in L; \quad L_k + L_m \in L, \forall L_k, L_m \in L;$$

$$\tilde{L}: \left\{\tilde{L}_1, \tilde{L}_2, \ldots, \tilde{L}_n\right\} \equiv L; \ \overline{L}: \left\{\overline{L}_1, \overline{L}_2, \ldots \overline{L}_n\right\} \equiv L; \ (L_i * L): \left\{L_i * L_1, L_i * L_2, \ldots, L_i * L_n\right\} \equiv L, \forall L_i \in L;$$

$$(L * L_i): \left\{L_1 * L_i, L_2 * L_i, \ldots, L_n * L_i\right\} \equiv L, \forall L_i \in L; \qquad (L_i + L): \left\{L_i + L_1, L_i + L_2, \ldots, L_i + L_n\right\} \equiv L, \forall L_i \in L.$$

In the following text we will consider the *L-set* as semantically closed one.

3.1.7. Operations with semantic constants

In the semantically closed L-set there should be relations *DIF, NIL* and *SAME* as they were defined in 3.1.1.

This means that at least one of possible relations should take place between a pair of objects and at each object has at least one property. The negation operation defines the main relationship between the two main constants as follows:

$$\overline{DIF} \equiv \overline{SAME} \equiv NIL.$$

Properties of DIF:

- neutrality to semantic sum $\quad DIF + L_k \equiv L_k$.

Proof: $\quad (P(A_i, DIF + L_k, A_j) \Leftrightarrow \underbrace{P(A_i, DIF, A_j)}_{true} \wedge P(A_i, L_k, A_j) \Leftrightarrow P(A_i, L_k, A_j)) \Leftrightarrow DIF + L_k \equiv L_k;$

- elimination in semantic multiplication $\quad DIF * L_k \equiv L_k * DIF \equiv DIF;$
- inversion of *DIF* $\quad \tilde{}(DIF) \equiv DIF;$

Properties of SAME:

- inversion of *SAME* $\quad \tilde{}(SAME) \equiv SAME;$
- neutrality to semantic sum $\quad SAME + L_k \equiv L_k;$

Properties of $SAME^b$:

- inversion of $SAME^b$ $\quad \tilde{}(SAME^b) \equiv SAME^b;$
- neutrality to semantic sum $\quad SAME^b + L_k \equiv L_k;$
- neutrality to semantic multiplication $\quad L_k * SAME^b \equiv SAME^b * L_k \equiv L_k.$
- annihilation $\quad$ a) $L_k + \tilde{L}_k \equiv \begin{cases} SAME^b, \textit{ for relations}; \\ L_k, \textit{ for properties.} \end{cases}$

  b) $L_k * \tilde{L}_k \equiv \tilde{L}_k * L_k \equiv SAME^b.$

- elimination (left) $\quad L_k + L_k * L_m \equiv L_k * L_m.$

  Proof: $\quad L_k + L_k * L_m \equiv L_k * SAME^b + L_k * L_m \equiv L_k * (SAME^b + L_m) \equiv L_k * L_m;$

- elimination (right) $L_k + L_m * L_k \equiv L_m * L_k$.

3.2. Semantic equations

An equation of the Algebra (semantic equation) is the equality of two expressions which are include only semantic relations' names as operands and only semantic operations with these operands. In semantic equation there should be at least one unknown operand. We use the following general expression to denote an equation: $F_1(L^1) = F_2(L^2)$, where F_1 , F_2 are functions, which use operations of the Algebra and they have known operands respectively subsets L^1, L^2 of the semantically closed L-set. Here and later we use "=" in semantic equations to tell the difference between semantic equations and semantic identities where we use "$\equiv$". We will consider equations with one unknown operand L_x, which is defined on the whole L-set.

The solution of such equation is the relation L_w, which, being substituted in an equation instead of the operand L_x will transform it to an identity. We will use notation $(equation)^{L_x = L_w}$ as a statement that relation L_w is the solution of the *equation*.

To solve semantic equations the following rules are supposed to be used:

$$(F_1(L^1) = F_2(L^2))^{L_x = L_w} \Leftrightarrow (\tilde{F}_1(L^1) = \tilde{F}_2(L^2))^{L_x = L_w};$$

$$(F_1(L^1) = F_2(L^2))^{L_x = L_w} \Leftrightarrow (\overline{F}_1(L^1) = \overline{F}_2(L^2))^{L_x = L_w};$$

$$(F_1(L^1) = F_2(L^2))^{L_x = L_w} \Leftrightarrow (F_1(L^1) + L_k = F_2(L^2) + L_k)^{L_x = L_w}, \forall L_k \in L;$$

$$(F_1(L^1) = F_2(L^2))^{L_x = L_w} \Leftrightarrow (F_1(L^1) * L_k = F_2(L^2) * L_k)^{L_x = L_w}, \forall (L_k \neq DIF \in L);$$

$$(F_1(L^1) = F_2(L^2))^{L_x = L_w} \Leftrightarrow (L_k * F_1(L^1) = L_k * F_2(L^2))^{L_x = L_w}, \forall (L_k \neq DIF \in L).$$

Using the above rules and properties of semantic operations, we present methods of the decision of following five types of equations of the Algebra.

$$\tilde{L}_x = L_i \Rightarrow \tilde{\tilde{L}}_x = \tilde{L}_i \Rightarrow L_x = \tilde{L}_i; \qquad \overline{L}_x = L_i \Rightarrow \overline{\overline{L}}_x = \overline{L}_i \Rightarrow L_x = \overline{L}_i;$$

$$L_x + L_i = L_j \Rightarrow L_x + L_i + \tilde{L}_i = L_j + \tilde{L}_i \Rightarrow L_x + SAME^b = L_j + \tilde{L}_i \Rightarrow L_x = L_j + \tilde{L}_i;$$

$$L_x * L_i = L_j \Rightarrow L_x * L_i * \tilde{L}_i = L_j * \tilde{L}_i \Rightarrow L_x * SAME^b = L_j * \tilde{L}_i \Rightarrow L_x = L_j * \tilde{L}_i;$$

$$L_i * L_x = L_j \Rightarrow \tilde{L}_i * L_i * L_x = \tilde{L}_i * L_j \Rightarrow SAME^b * L_x = \tilde{L}_i * L_j \Rightarrow L_x = \tilde{L}_i * L_j.$$

4. An Algebra with Contexts in a Semantic Metanetwork

In this chapter we will develop the Algebra, described in the previous chapter, by adding a new semantic operation which allows to reason with multilevel structure of contexts in a semantic metanetwork.

4.1. Operation of semantic interpretation

Semantic interpretation operation makes it possible to take into account the properties of the appropriate context when one defines knowledge of a relationship between a pair of objects. If we have knowledge L_k about the name of relation between the objects A_i and A_j, and knowledge L'_n about the name of property of the context A'_l of this relation (as shown in Figure 2a), then we can obtain a new knowledge about this relation which is supposed to be an interpretation of the first knowledge in the context of the second one. By the same way it is possible to interpret knowledge about a property of an object in certain context as it is shown in Figure 2b.

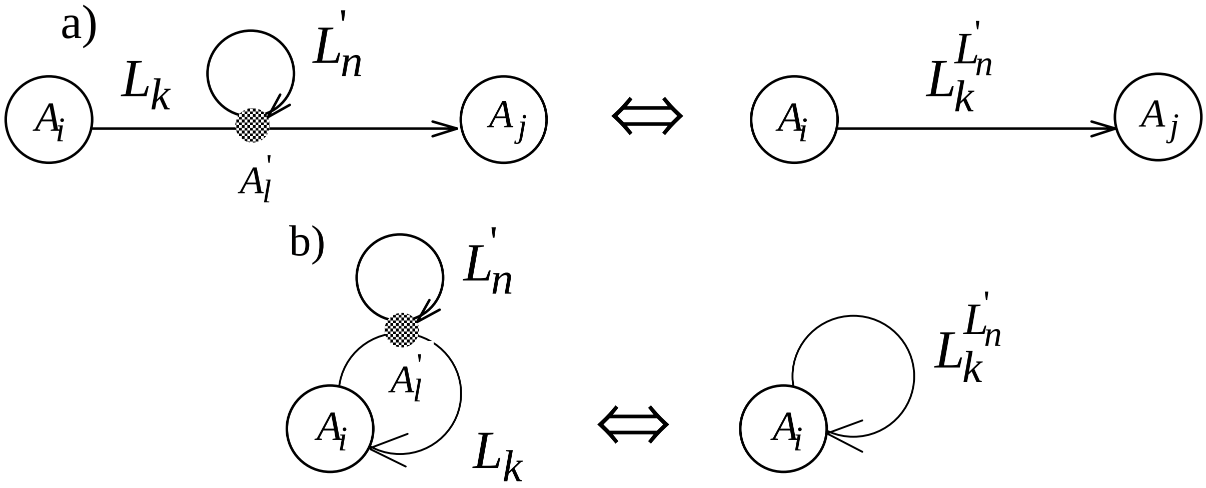

Fig. 2. Semantic interpretation

Formally: $P(A_i, L_k, A_j) \wedge P(A_l', L_n', A_l') \wedge ist(A_l', P(A_i, L_k, A_j)) \Leftrightarrow P(A_i, L_k^{L_n'}, A_j)$,

where $L_k^{L_n'}$ denotes the interpretation of the knowledge L_k using the knowledge L'_n about the context A_l'. This can be named and included to the set *L*. We use a property of a context in the description of our basic operation which connects different levels of a semantic metanetwork assuming that different contexts with the same properties make the same effect to the interpretation of knowledge.

We expand the definition of the relations *SAME* and $SAME^b$, partly defined in the previous chapter, adding the following properties with the new operation:

1. $L_k^{SAME} \equiv L_k$.

This means an abstract case when the context of a relation contains only the universal property *SAME*. In such case the context cannot change an interpretation of this relation;

2. $SAME^{L_k} \equiv L_k$.

This means also an abstract case when the knowledge being interpreted contains only the universal property *SAME*. In such case the only knowledge obtained as a result of interpretation is the knowledge about context;

3. $(SAME^b)^{L_k} \equiv L_k^b$.

This means that interpretation of the equivalence relation between two objects inherits the properties of context L_k producing its binary equivalent L_k^b;

4. $(L_k^{L_m} \equiv L_n) \Leftrightarrow (L_n^{\overline{L}_m} \equiv L_k)$ - axiom of extracting knowledge.

This means that if some knowledge L_n has been obtained as interaction of knowledge L_k and property L_m of some context, then removal of that property from the context of the interpretation result leads to the restoration of initial knowledge;

Consequence: $L_k^{\overline{L}_k} \equiv SAME^{[b]}$;

5. $(L_k^{L_m} \equiv L_n) \Leftrightarrow (\overline{L}_k^{L_n} \equiv L_m)$ - axiom of extracting context.

This means that if some knowledge L_n has been obtained as interaction of knowledge L_k and property L_m of some context, then removal of the initial knowledge L_k in the context of the interpretation result leads to the restoration of initial context.

Consequence: $\overline{L}_k^{L_k} \equiv SAME^{[b]}$;

Other properties of the semantic interpretation operation will be considered in the next chapter as transformations with contexts.

4.2. Transformations with contexts in the Algebra

In this paragraph, we describe the rules of transformation of the Algebra expressions which include contexts.

Inversion in the context. The inverse result of interpretation a relation in a context is equal to the result of interpretation the inverse relation in the same context. Formally:

$$^{\sim}(L_k^{L_m}) \equiv \tilde{L}_k^{L_m} .$$

Negation in the context. The negative result of interpretation a relation in a context is equal to the result of interpretation the negative relation in the same context, and also it equals to the result of interpretation of this relation in a negative context. Formally:

$$\overline{L_k^{L_m}} \equiv \overline{L}_k^{L_m} \equiv L_k^{\overline{L_m}} .$$

Addition in the context. The result of interpretation of the sum of two not-conflicting relations in a context is equal to the semantic sum of these two relations interpreted separately in the same context. Formally:

$$(L_k + L_m)^{L_n} \equiv L_k^{L_n} + L_m^{L_n}, \ \ when\ L_k \neq \overline{L}_m.$$

Multiplication in the context. The result of interpretation of the semantic multiplication of two not-conflicting relations in a context is equal to the semantic multiplication of these two relations interpreted separately in the same context. Formally:

$$(L_k * L_m)^{L_n} \equiv L_k^{L_n} * L_m^{L_n}, \ \ when\ L_k \neq \overline{L}_m .$$

Interpretation in the sum of contexts. The result of interpretation of a relation in a semantic sum of two not-conflicting contexts is equal to the semantic sum of this relation interpreted separately in these two contexts. Formally:

$$L_k^{L_m + L_n} \equiv L_k^{L_m} + L_k^{L_n}, \ \ when\ L_m \neq \overline{L}_n .$$

Addition of interpreted relations. The result of a semantic sum of two not-conflicting relations interpreted separately in two not-conflicting contexts is equal to the semantic sum of these relations interpreted in the semantic sum of these contexts:

$$L_k^{L_m} + L_r^{L_n} \Rightarrow (L_k + L_r)^{L_m + L_n}, \ \ when\ L_k \neq \overline{L}_r,\ L_m \neq \overline{L}_n .$$

Multiplication of interpreted relations. The result of a semantic multiplication of two not-conflicting relations interpreted separately in two not-conflicting contexts is equal to the semantic multiplication of these relations interpreted in the semantic sum of these contexts:

$$L_k^{L_m} * L_r^{L_n} \Rightarrow (L_k * L_r)^{L_m + L_n}, \ \ when\ L_k \neq \overline{L}_r,\ L_m \neq \overline{L}_n .$$

Multilevel interpretation. The result of a relation interpretation in several levels of contexts does not depend on the order of interpretation:

$$(L_k^{L_m})^{L_n} \equiv L_k^{(L_m^{L_n})},\ \ when\ L_m \neq \overline{L}_k,\ L_m \neq \overline{L}_n.$$

4.3. Equations with contexts in the Algebra

To solve equations with the operation of semantic interpretation, the following main rules should be used:

$$(F_1(L^1) = F_2(L^2))^{L_x = L_w} \Leftrightarrow (F_1(L^1)^{L_k} = F_2(L^2)^{L_k})^{L_x = L_w}, \forall L_k \in L;$$

$$(F_1(L^1) = F_2(L^2))^{L_x = L_w} \Leftrightarrow (L_k^{F_1(L^1)} = L_k^{F_2(L^2)})^{L_x = L_w}, \forall L_k \in L\ .$$

Using the above properties and rules, we present solution methods of the following two types of equations of the Algebra.

$$L_x^{L_i} = L_j \Leftrightarrow (L_x^{L_i})^{\overline{L}_i} = L_j^{\overline{L}_i} \Leftrightarrow L_x^{(L_i^{\overline{L}_i})} = L_j^{\overline{L}_i} \Leftrightarrow L_x^{SAME} = L_j^{\overline{L}_i} \Leftrightarrow L_x = L_j^{\overline{L}_i};$$

$$L_i^{L_x} = L_j \Leftrightarrow \overline{L_i^{(L_i^{L_x})}} = \overline{L_i^{L_j}} \Leftrightarrow (\overline{L_i^{L_i}})^{L_x} = \overline{L_i^{L_j}} \Leftrightarrow SAME^{L_x} = \overline{L_i^{L_j}} \Leftrightarrow L_x = \overline{L_i^{L_j}}.$$

Using above samples, one can solve more complex equations, for example:

$$\tilde{L}_1 * L_2^{(\tilde{L}_3 * \tilde{L}_x * L_4 + L_5)^{\tilde{L}_6}} * L_7 + L_8 = L_9 \Leftrightarrow \tilde{L}_1 * L_2^{(\tilde{L}_3 * \tilde{L}_x * L_4 + L_5)^{\tilde{L}_6}} * L_7 = L_9 + \tilde{L}_8 \Leftrightarrow$$

$$\Leftrightarrow L_2^{(\tilde{L}_3 * \tilde{L}_x * L_4 + L_5)^{\tilde{L}_6}} * L_7 = \tilde{\tilde{L}}_1 * (L_9 + \tilde{L}_8) \Leftrightarrow L_2^{(\tilde{L}_3 * \tilde{L}_x * L_4 + L_5)^{\tilde{L}_6}} = L_1 * (L_9 + \tilde{L}_8) * \tilde{L}_7 \Leftrightarrow$$

$$\Leftrightarrow L_2^{(\tilde{L}_3 * \tilde{L}_x * L_4 + L_5)} = (L_1 * (L_9 + \tilde{L}_8) * \tilde{L}_7)^{\overline{\tilde{L}}_6} \Leftrightarrow \tilde{L}_3 * \tilde{L}_x * L_4 + L_5 = \overline{L}_2^{(L_1 * (L_9 + \tilde{L}_8) * \tilde{L}_7)^{\overline{\tilde{L}}_6}} \Leftrightarrow$$

$$\Leftrightarrow \tilde{L}_3 * \tilde{L}_x * L_4 = \overline{L}_2^{(L_1 * (L_9 + \tilde{L}_8) * \tilde{L}_7)^{\overline{\tilde{L}}_6}} + \tilde{L}_5 \Leftrightarrow \tilde{L}_x * L_4 = \tilde{\tilde{L}}_3 * (\overline{L}_2^{(L_1 * (L_9 + \tilde{L}_8) * \tilde{L}_7)^{\overline{\tilde{L}}_6}} + \tilde{L}_5) \Leftrightarrow$$

$$\Leftrightarrow \tilde{L}_x = L_3 * (\overline{L}_2^{(L_1 * (L_9 + \tilde{L}_8) * \tilde{L}_7)^{\overline{\tilde{L}}_6}} + \tilde{L}_5) * \tilde{L}_4 \Leftrightarrow L_x = \sim(L_3 * (\overline{L}_2^{(L_1 * (L_9 + \tilde{L}_8) * \tilde{L}_7)^{\overline{\tilde{L}}_6}} + \tilde{L}_5) * \tilde{L}_4).$$

5. Reasoning with a Semantic Metanetwork

Using semantic metanetworks and equations of the Algebra, one can solve the four reasoning problems with contexts described in the following subsections.

Deriving an interpreted knowledge (decontextualization). Deriving of interpreted knowledge means here deriving the formula for knowledge interpreted using all known levels of its context. This problem is solved using operations of the Algebra.

We define the knowledge of a relation $L_{A_i - A_j}$ between any pairs of objects (A_j, A_k) from the same level of a semantic metanetwork as a semantic sum over all possible paths between these objects (A_j, A_k) that exist at this level of the metanetwork.

We define the knowledge L_{A_i} of an object A_i of a semantic metanetwork in one level as a semantic sum over all knowledge of the relations that connect this object with all objects of the same level, including the object itself:

$$L_{A_i} = \sum_j L_{A_i - A_j}.$$

The interpreted knowledge of any relation, considering all contexts and metacontexts, is derived by the following schema:

$$< \text{interpreted knowledge} > = < \text{knowledge} >^{< \text{knowl. about context} >^{\cdots^{< \text{knowl. about metacontext of } n - \text{th level} >}}}.$$

As an example let us derive the interpreted knowledge of the relation between A_1 and A_3 in Figure 1. We will start from the top level of the metanetwork and define knowledge about metacontexts A_1'', A_2'', A_3'':

$$L_{A_1''} = L_{A_1''-A_2''} + L_{A_1''-A_3''} + L_{A_1''-A_1''} = L_1'' + L_1'' * L_2'' + SAME = L_1'' * (SAME + L_2'') = L_1'' * L_2'';$$

$$L_{A_2''} = \tilde{L}_1''' + L_2''; \qquad L_{A_3''} = \tilde{L}_2''' * \tilde{L}_1'''.$$

Now we can continue at the first level of the metanetwork and derive the interpreted knowledge of the first level relations:

$$L_{A_1'-A_2'} = (L_1')^{L_{A_1''}} = (L_1')^{L_1'' * L_2''};$$

$$L_{A_1'-A_3'} = (L_1')^{L_{A_1''}} * (L_2')^{L_{A_2''}} = (L_1')^{L_1'' * L_2''} * (L_2')^{\tilde{L}_1''' + L_2''} = (L_1' * L_2')^{L_1'' * L_2'' + \tilde{L}_1'''};$$

$$L_{A_1'-A_4'} = (L_1')^{L_{A_1''}} * (L_3')^{L_{A_3''}} = (L_1')^{L_1'' * L_2''} * (L_3')^{\tilde{L}_2''' * \tilde{L}_1'''};$$

$$\ldots$$

$$L_{A_4'-A_3'} = (\tilde{L}_3')^{L_{A_3''}} * (L_2')^{L_{A_2''}} = (\tilde{L}_3')^{\tilde{L}_2''' * \tilde{L}_1'''} * (L_2')^{\tilde{L}_1''' + L_2''} = (\tilde{L}_3' * L_2')^{\tilde{L}_2''' * \tilde{L}_1''' + L_2''}.$$

The knowledge about contexts A_1', A_2', A_3' of the first level is derived as follows:

$$L_{A_1'} = (L_1' * L_2')^{L_1'' * L_2'' + \tilde{L}_1'''} + (L_1' * L_3')^{L_1'' * L_2'' + \tilde{L}_2''' * \tilde{L}_1'''} = (L_1' * (L_2' + L_3'))^{L_1'' * L_2'' + \tilde{L}_2''' * \tilde{L}_1'''};$$

$$L_{A_2'} = (\tilde{L}_1')^{L_1'' * L_2''} + (L_2')^{\tilde{L}_1''' + L_2''} + (L_3')^{\tilde{L}_2''' * \tilde{L}_1'''} = (\tilde{L}_1' + L_2' + L_3')^{L_1'' * L_2'' + \tilde{L}_2''' * \tilde{L}_1'''};$$

$$L_{A_3'} = (\tilde{L}_2' * \tilde{L}_1')^{L_1'' * L_2'' + \tilde{L}_1'''} + (\tilde{L}_2' * L_3')^{L_2'' + \tilde{L}_2''' * \tilde{L}_1'''} = (\tilde{L}_2' * (\tilde{L}_1' + L_3'))^{L_1'' * L_2'' + \tilde{L}_2''' * \tilde{L}_1'''}.$$

Now it is possible to derive the interpreted knowledge about the relation between A_1 and A_3 taking all contexts and metacontexts into account as:

$$L_{A_1-A_3} = (L_1)^{L_{A_1'}} * (L_2)^{L_{A_2'}} + (L_3)^{L_{A_3'}} =$$

$$= (L_1 * L_2 + L_3)^{(L_1' * L_2' + L_1' * L_3' + \tilde{L}_2' * \tilde{L}_1' + \tilde{L}_2' * L_3')^{(L_1'' * L_2'' + \tilde{L}_2'' * \tilde{L}_1'')}}.$$

How to interpret formulas of the Algebra? The application area in such a problem defines the way of representing context and metacontexts. If the internal structure of a context is known, one can, for example, use the formalism of “semantic balance” between internal structure of a context and its external relationships [13]. Some examples of interpretation technique in the domain of temporal relations are given in [3]. The contexts of a relationship in [18, 26] are rules that define conditions of appearing and disappearing of relations. The knowledge about knowledge sources can also be considered as a context for knowledge base refinement as in [20, 21]. Farquhar et al. [12] have used contexts to integrate databases, and Halpern and Moses [15] have used contexts to reason about knowledge and belief of multiple agents. In [19], the knowledge about the relationship of an expert with his colleagues is used as a context to interpret knowledge acquired from this expert.

Deriving unknown knowledge that is interpreted when the result of interpretation and the context of interpretation are known (contextualization). This problem occurs when some knowledge has been interpreted in some context and we have all knowledge about this context and knowledge that is the result of interpretation. For example, let us suppose that your colleague, whose context you know well, has described you a situation. You use knowledge about context of this person to interpret the “real” situation. Example is more complicated if several persons describe you the same situation. In this case, the context of the situation is the semantic sum over all personal contexts.

This second reasoning problem can be solved using the following equation:

$$L_x^{<\text{knowledge about context}>} = <\text{interpreted knowledge}>;$$

Deriving unknown context of interpretation when the knowledge and its interpretation in this context are known (context recognition). This problem occurs when we have knowledge that has been interpreted in some unknown context and we also know what is the result of interpretation. For example let us supposed that someone sends you a message describing the situation that you know well. You compare your own knowledge with the knowledge you received. Usually you can derive your opinion about the sender of this letter. Knowledge about the source of the message, you derived, can be considered as certain context in which real situation has been interpreted and sometimes it can help you to recognize a source or at least his motivation to change the reality.

This third reasoning problem can be solved using the following equation:

$$<\text{knowledge}>^{L_x} = <\text{interpreted knowledge}>.$$

Lifting (relative decontextualization). This means deriving knowledge interpreted in some context if it is known how this knowledge was interpreted in another context. This problem is solved by successive solution of the above contextualization and decontextualization problems. Let L_k is the result of interpretation of some knowledge L_x in the context L_m. The problem is to derive how this knowledge would be interpreted in the context L_n. Thus we have the following procedure of lifting:

$$(L_x^{L_m} = L_k) \Leftrightarrow (L_x = L_k^{\overline{L}_m})[contextualization] \Leftrightarrow (L_x^{L_n} = L_k^{\overline{L}_m^{L_n}})[decontextualization].$$

The formal tools, that are necessary to handle these reasoning problems, are presented in chapters 3 and 4 of this paper.

6. Conclusion

In our previous papers, we have described metaobjects in a metanetwork as rules, determining behavior of relations. In this paper, we propose interpretation of metaobjects as contexts. This enables ordering of contexts into multilevel representation, that can be used during reasoning process. We have presented a general framework for solving four types of problems: how to derive interpretation of knowledge in a context; how to derive knowledge that was interpreted; how to derive knowledge about context of interpretation and how to change knowledge from one context to another.

One way was shown how to interpret expressions of the Algebra. Further research is needed to make more universal tools for interpreting such expressions in various applications where it is reasonable to consider several levels of context. It is also planned to include temporal component to the multilevel knowledge representation by considering dynamically changed contexts. Another important problem is how to use context if incomplete and inconsistent knowledge about it is acquired from several knowledge sources. It is also necessary to consider cases when inconsistent and incomplete knowledge of multiple experts is interpreted using inconsistent and incomplete knowledge about a context.

References

1. V. Akman, M. Surav: Steps Towards Formalizing Context. AI Magazine, V.17, No.3, 1996, pp. 55-72.
2. G. Attardi, M. Simi: A Formalization of Viewpoints. Fundamenta Informaticae, V.23, Ns.2-4, 1995, pp. 149-174.
3. M.F. Bondarenko, V.A. Grebenyuk, V.Ya. Terziyan: Reasoning Based on the Algebra of Semantic Relation. Pattern Recognition and Image Analysis, V. 3, No. 4, Interperiodica Publisher 1993, pp. 488-499.
4. P. Brezillon: Context Needs in Cooperative Building of Explanations. In: Proc. of the First European Conference on Cognitive Science in Industry, 1994, pp. 443-450.
5. P. Brezillon, S. Abu-Hakima: Using Knowledge in its Context: Report on the IJCAI-93 Workshop. AI Magazine, V.16, No. 1, 1995, pp. 87-91.
6. P. Brezillon, E. Cases: Cooperating for Assisting Intelligently Operators. In: Proc. of Actes International Workshop on the Design of Cooperative Systems, 1995, pp. 370-384.
7. S. Buvac, V. Buvac, I.A. Mason: Metamathematics of Contexts. Fundamenta Informaticae, V.23, No.2, 1995, pp. 263-301.
8. S. Buvac, V. Buvac, I.A. Mason: Propositional Logic of Context. In: Proceedings of the Eleventh AAAI Conference, 1993, Washington DC, pp. 412-419.
9. S. Buvac, V. Buvac, I.A. Mason: Sematics of Propositional Contexts. In: Proceedings of the Eight International Symposium on Methodologies for Intelligent Systems, Springer Verlag, Lecture Notes in Artificial Intelligence, V.869, 1994, pp. 468-477.
10. S. Buvac: Quantificational Logic of Context. In: Proceedings of the Thirteenth AAAI Conference, 1996, Menlo Park, California.
11. B. Edmonds: A Simple-Minded Network Model with Context-like Objects. CPM Report 97-15, The workshop on Context at the European Conference on Cognitive Science (ECCS'97), Manchester, April, 1997.
12. A. Farquhar, A. Dappert, R. Fikes, W. Pratt: Integrating Information Sources Using Context Logic. Technical Report KSL-95-12, Stanford.
13. V. Grebenyuk, H.B. Kaikova, V.Ya. Terziyan, S. Puuronen: The Law of Semantic Balance and its Use in Modeling Possible Worlds. In: STeP-96 - Genes, Nets and Symbols, Publications of the Finnish AI Society, Vaasa, Finland, 1996, pp. 97-103.
14. R.V. Guha: Contexts: A Formalization and Some Applications, Stanford Ph.D. Thesis, 1991.
15. J.Y. Halpern, Y. Moses: A Guide to Completeness and Complexity for Modal Logic of Knowledge and Belief. Artificial Intelligence, V.54, 1992.
16. J. McCarthy: A Logical AI Approach to Context. Technical Note, Computer Science Department, Stanford University, 1995, Available in http://www-formal.stanford.edu/jmc/index.html.
17. J. McCarthy: Notes on Formalizing Context. In: Proc. of 13 International Joint Conference on Artificial Intelligence, 1993, pp. 555-560.
18. S. Puuronen, V.Ya. Terziyan: A Metasemantic Network. In: New Directions in Artificial Intelligence: Publications of the Finnish AI Society, 1992, pp. 136-143.
19. S. Puuronen, V.Ya. Terziyan: Colleague-Oriented Interpretation of Knowledge Acquired from Multiple Experts, In Patterson, D., Leedham, G., Warendorf, K., and Hwee, T.A. (Eds.), *Proceedings of the Joint 1997 Pacific Asian Conference on Expert Systems/Singapore International Conference on Intelligent Systems (PACES/SPICIS 97),* The Nanyang Technological University, 1997, pp. 737-741.
20. S. Puuronen, V.Ya. Terziyan: Modeling Consensus Knowledge from Multiple Sources Based on Semantics of Concepts, In: Challenges of Design, ER'96 International Conference on Conceptual Modeling, Cottbus, Germany, October 1996 , pp. 133-146.
21. S. Puuronen, V.Ya. Terziyan: Voting-Type Technique of the Multiple Expert Knowledge Refinement, In: Sprague, R. H., (Ed.), *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, Vol V, IEEE Computer Society Press, 1997, pp. 287-296.
22. S. Russel, E. Wefald: Principles of Metareasoning. Artificial Intelligence, V. 49, Nos 1-3, 1991, pp. 361-395.
23. L. Shastri: Default Reasoning in Semantic Networks: A Formalization of Recognition and Inheritance. Artificial Intelligence, V. 39, No. 3, 1989, pp. 283-355.
24. S.C. Shapiro: Cables, Paths and "Subconscious" Reasoning in Propositional Semantic Networks. In: J.F. Sowa (ed.): Principles of Semantic Networks, Morgan Kaufmann 1991, pp. 137-156.
25. W.A. Taylor, D.H. Weimann, P.J. Martin: Knowledge Acquisition and Synthesis in a Multiple Source Multiple Domain Process Context. Expert Systems with Applications, V.8, No.2, 1995, pp. 295-302.
26. V.Ya. Terziyan: Multilevel Models for Knowledge Bases Control and Their Applications to Automated Information Systems. Post Doctoral Degree Thesis, State Technical University of Radioelectronics, Kharkov, Ukraine, 1993, (in Russian).
27. W.A. Woods: Understanding Subsumption and Taxonomy: A Framework for Progress. In: J.F. Sowa (ed.): Principles of Semantic Networks, Morgan Kaufmann 1991, pp. 45-94.