Stepping Theories of Active Logic with Two Kinds of Negation

Mikhail M. VINKOV[1], *Igor B. FOMINYKH*[2]

[1]Department of Robotics and Complex Automation, Faculty of Robotics and Cybernetics,
Bauman Moscow State Technical University, 2nd Baumanskaya street 5, 105005 Moscow, Russia
[2]Department of Applied Mathematics, Institute of Automatics and Computer Engineering,
Moscow Power Engineering Institute, Krasnokazarmennaya 14, 111250 Moscow, Russia

vinkovmm@mail.ru, igborfomin@mail.ru

DOI: 10.15598/aeee.v15i1.1990

Abstract. *This paper formulates a stepping theory formalism with two kinds of negation dealing with one of the areas of Active Logic, a new kind of logic aimed at performing practical tasks in real time knowledge-based AI systems. In addition to the standard logical negation, the proposed formalism uses the so-called subjective negation interpreted as inability to arrive at some conclusion through reasoning by a current time. The semantics of the proposed formalism is defined as an argumentation structure.*

Keywords

AI systems, logical negation, subjective negation.

1. Introduction

Active Logic is a conceptual system, the principles of which are met by metareasoning formalisms with a high degree of tolerance for contradictions that can link results of reasoning with specific points in time. This makes Active Logic especially suited for use in real time knowledge-based AI systems that can be used in management and process control solutions, that need to operate under strict time constraints. One area of Active Logic deals with the so-called stepping theory systems that combine the principles of Active Logic with those of logic programming. In this work we propose a system of stepping theories that, in addition to the standard negation (strong negation), employs the so-called subjective negation, which is interpreted as inability/failure to work out a logic formula by the current point in time. This approach makes control process under strict time constraints more effective compared to other stepping theory formalisms. The semantic of stepping theory formalism with two kinds of negation is defined as an argumentation structure. Time, as is the case in some of the other stepping logic systems, is interpreted as an external entity that does not depend on the internal structure of the set of rules applied by the stepping theory.

2. Metareasoning and Active Logic

2.1. Task Solving in Hard Real Time and Need for Metareasoning

One of the most difficult problems that has to be tackled by knowledge based AI systems operating under the conditions of the so-called 'hard time' is how to manage the reasoning process of a cognitive agent (hereinafter, the agent) that is solving task by means of reasoning on the basis of its knowledge and its observations of the environment. Operating under these conditions means that there is a certain critical time point or deadline by which the agent has to solve its current task [1]. If the task has not been accomplished by deadline, the consequences can be catastrophic and thus not meeting the deadline for the agent is unacceptable, and thereafter the agent's actions become completely meaningless. One example of a situation where task solving has to be done under hard real time conditions is choosing what kind of resuscitation to perform on a patient whose heart and breathing have just stopped. You cannot wait too long to decide what to do: in five minutes irreversible changes will start happening in the patient's body beyond saving. Naturally, agents operating under such conditions must have different priorities than agents functioning under soft real time conditions, where there is no deadline at all or even if the deadline is not met, the agent's performance quality will deteriorate in proportion to the amount of time delayed. Obviously, the cognitive process of an agent acting under hard real time conditions must be closely monitored by the agent at all times. The main function of such close monitoring is to identify unforeseen difficulties (anomalies) that prevent the agent from achieving its target. This means that such an agent needs to be aware not just of its environment and the task it has to perform but also of its own cognitive processes, in other words, this agent has to be able to engage not only in reasoning but in metareasoning or reasoning about reasoning. It's clear that metareasoning capabilities are a must for intelligent agents operating under hard real time conditions, because it's only agents that are capable of analyzing their own cognitive processes that can adequately tackle the threats posed by anomalies.

2.2. Logical Omniscience Problem

Wherever there is reasoning, including metareasoning, there must be logic, but not all logic systems, including epistemic logic systems that specialize in formalizing the knowledge and beliefs the agent uses for reasoning, fit the bill if reasoning tasks must be completed under hard time constraints. The vast majority of logic systems developed to formalize the knowledge and beliefs of agents and that use the modal approach are implicitly based on the assumption that the reasoning performed by the agent happens instantaneously and the results and conclusions are made available immediately or on demand. There are situations in which such an assumption is perfectly justified. For example, an agent can only use very simple reasoning while operating under conditions that don't impose critical time constraints. However, in situations where the agent is operating under strict time constraints, the assumption that reasoning happens instantaneously becomes one assumption too far. It leads to the so-called logical omniscience problem, which is very well known in epistemic logic. Consequently, agents that are subject to this problem are known as omniscient or ideal agents [2] and [3]. It goes without saying that an ideal agent does not need to care about how much time it's got left because it completes all reasoning tasks instantaneously. However, if we can't assume omniscience for our agent, then a way must be found to deal with the logical omniscience problem. Solutions to this problem have been proposed both within the modal approach (for example, see [4]) and outside of it [5]. As was shown in [6] the solutions found within the classical modal approach all share one principal drawback. None of them can be used to arrive at the following conclusion: there is a risk that agent i won't be able to derive formula F within time constraint t. It has to be noted that the results of this kind are extremely important in situations when there are hard constraints on the amount of time that agent i has at its disposal. Outside the modal approach this drawback is successfully dealt with in logical systems that are described as the so-called 'Active Logics' [7] or systems similar to them [8], in which the agent's reasoning is interpreted not as a sequence of formulas (statements) regarded as a whole and existing outside of time, but rather, as a process happening in time. Below, we talk about logic systems that fall within the concept of Active Logic.

2.3. Fundamentals of Active Logics

Active logic conceptually combines a number of formalisms or reasoning that is said to be 'situated in time'. Logic systems that meet this criterion can be based on the combination of the following three basic principles, depending on their complexity.

1) Keeping Track of Time

In active logic the reasoning process is situated in time and implies sequential performance of deductive cycles, which we're going to refer to as inference steps. Time's kept track of using the special unary predicate $now(.)$. The following inference rule applies to this predicate:

$$\frac{t : now(t)}{t+1 : now(t+1)}. \tag{1}$$

It should be noted that the formula $now(t)$ does not get inherited at the point in time (inference step) $t+1$, as is sometimes the case with 'ordinary' formulae.

2) Self-Awareness

This principle implies that the agent is aware of what it knows at this point in time as well as of what they don't know at this point in time. In the most general form this principle is implemented using the following inference rules:

$$\frac{t : \varphi}{t+1 : K(t,\varphi)}, \tag{2}$$

$$\frac{t : \varphi, sub(\phi,\varphi), [\phi]}{t+1 : \neg K(t,\phi)}, \tag{3}$$

where ϕ is any formula the agent does not know at time t, but which can be inferred from formula φ that the agent already knows about, in other words, it's a sub-formula whose existence the agent is *aware of*, $sub(.,.)$ is a binary meta-predicate that expresses the relation 'to be a sub-formula of, $[\phi]$ is a notation that means that formula ϕ is absent from the agent's current knowledge base at time t. $K(.,.)$ is a binary meta-predicate (and not a modal operator!), which expresses the fact that the agent knows some formula at some time point.

Equation (2) and Eq. (3) are used in order to compare and contrast the knowledge the agent possesses at this point in time with the expectations that the agent had with regards to the inference process it is performing. If these expectations contradict reality, this means the agent may be faced with an anomaly and thus needs to take measures to deal with its consequences.

3) Identification and Elimination of Direct Contradictions

In situations the available time is limited, contradictions are quite natural and it's of vital importance that when faced with contradictory information, instead of crashing or throwing an error, the agent keep going and use this contradictory information to adjust its inference process. Some active logic systems deal with the problem of contradictory information through a mechanism that allows them to identify and eliminate the so-called direct contradictions, i.e. pares of formulae in which one negates the other. Identification and subsequent processing of direct contradictions in the most general form can be achieved using the following inference rule:

$$\frac{t : \varphi, \neg\phi}{next\ t : contra(t, \varphi, \neg\varphi)}, \tag{4}$$

where $contra(.,.,.)$ is a special ternary meta-predicate that returns true if at time t the current knowledge of the agent contains formulae ϕ and $\neg\phi$ (from here on out for any literal such as q we will use the symbol ¬q to refer to its complement). It should be noted that until very recently there was no satisfactory definition of the declarative semantic of Active Logic systems that use identification and elimination of direct contradictions [9] and [15].

2.4. Temporal Sensitivity and Time Granulation

As has already been stated above, the reasoning process in Active Logic systems is normally presented as a sequence of inference steps, each of which explicitly creates the so-called belief set, a finite set of formulae expressed in a language close to the language of first order logics. Each inference step is given a number interpreted as a point on a discrete linear timeline. In accordance with the principle of keeping track of time, this number is represented with the help of a special literal of the form $now(i)$, where i is the number of the inference step. This number can then be referenced in other formulae thereby allowing specific inference steps to be associated with other events happening within the system and in its environment. Add to that the self-awareness principle and our agent can not only establish whether a new formula appears in its belief set at a certain point of time but also whether a specific formula is absent from its belief set at a given point in time. Thus, the result of the inference process *now* depends on how soon a certain formula gets inferred or, to put it the other way, how long a certain formula remains absent from the current belief set of the agent. This property of active logic systems has come to be known as temporal sensibility.

One common drawback of most Active Logic systems is that they interpret time as an entity that is, in some sense, internal to these systems. And the flow of time is defined through the structure of inference rules used to infer new formulae from those that are already in the belief set. It is implicitly assumed that a unit of time is the duration of one inference step. Each time an inference step is completed, the virtual internal clock ticks once. At the same time, another (also implicit) assumption is made, namely, that each inference step has the same duration or that the variation in the amount of time taken by each inference step is so negligible that it can be safely ignored. In reality, though, the duration of each inference step is affected by the changes happening in the composition and structure of the knowledge acquired as a result of the on-going inference process and observations about the external environment. In addition, the duration of inference steps can be affected by random factors such as power surges, malfunctions of the other systems etc. In effect, the assumption that each inference step has roughly the same duration is very similar to logical omniscience and just like logical omniscience it often clashes with reality.

The Ref. [10] proposes an approach in which time is interpreted as an external entity that has no links to the structure of the agent's knowledge or the speed the agent completes deductive cycles at. This modification allows us to do away with the unrealistic assumption about 'internal time' and can be applied to any Active Logic system. In Active Logic systems time is regarded as an infinite ever increasing series of natural numbers form set N. We will be referring to it as G_{ck} (global clock). However, it should be taken into account, that the main purpose of this kind of logical systems is to model the behavior of agents under various conditions (= runs). So for each run, its own slightly different clock (C_k) will be used that best describes the conditions of the run (the time granulation principle). The clock used each time the model is run is a finite or infinite strictly increasing sequence whose members are interpreted as points in time (on the global clock) at which inference steps are completed, for example $C_k = \langle 0, 3, 5, 7, 10, \ldots \rangle$. We'll use the symbol C_k^* to designate the set of all such points in time. Each tick of the clock used in the model's run, just like the tick of the virtual internal clock that we talked about earlier, corresponds to the completion of one specific inference step. But the number of the inference steps corresponds not to the point in time at which it completes but only to the number of that point in time on the clock used for the model's run. This means that by changing the clock used for the model's run we can imitate different conditions that the agent operates under, allowing us to better model such things as the increase in the duration of inference steps the agent goes through as the amount of the information available to the agent increases. Furthermore, different agents can be assigned different local clocks to model, for example, differences in how 'smart' each agent is (their performance) or the fact that they engage with the environment at different points in time. Here, however, we'll only be considering the single agent case for simplicity's sake. We're also going to need two functions: $clock(.)$ and $rank(.)$. The first one maps set $\mathbb{N}$ unto set C_k^*. The term $clock(n)$ is interpreted as a point in time that has the number n on the clock used in the model's run. The second function is the reverse of the first one, meaning that for any $t \in C_k^*$ $rank(t) = clock^{-1}(t)$, it returns the number of the point in time t on clock C_k used for the model's run. Then, if the previous reference step completes at time t, the next inference step will complete at point in time $clock\ (rank(t)+1)$. Here on out we will be following the conventions of [11] to designate the point in time after time t on clock C_k, namely, we will be referring to it as *next t*.

3. Stepping Theories in Active Logic

One field in Active Logics introduced in [12] deals with various systems of the so-called stepping theories. The relation between stepping theories and other Active Logic systems is roughly the same as the relation between logic programming and other non-monotonic systems of traditional logic. We will expand on this point below. A stepping theory is a set of rules that sometimes includes a binary preference relation. The sets of rules in step theories sometimes are considered to comprise two subsets: the set of strict rules and the set of plausible rules. Without sacrificing generality, we're now going to focus only on stepping theories whose preference relation is empty and that only have plausible rules defined as follows:

$$N : a_1 \wedge a_2 ... \wedge a_m \Rightarrow b, \tag{5}$$

where N is a string of symbols designating the name of the rule, b is a propositional literal, $a_1, \ldots, a_m$ are propositional literals or first order logic literals of the kind $later(j)$ or $\neg later(j)$, where j is a natural number.

These rules reflect the principle of negative introspection in the following interpretation: if function $a_1 \wedge a_2 ... \wedge a_m$ is true and at this inference step it's unknown if formula $-b$ (which is a literal) is plausible or not, then it can be assumed that formula b is true (it should be reminded that $-b$ means the complement of b). Wherever it's convenient, the antecedents of rules are regarded as sets of literals. If such a set is empty, then the rules are regarded as being equivalent to rules with the same consequents, but whose antecedents only have a single literal $later(clock(rank(0))$, i.e. the rule 'kicks in' at the very first point in time on the clock of this stepping theory. Thus, the system of stepping theories can be regarded as a variation on Active Logic that's based solely on rules and that meets the principle of logic programming, according to which sets of literals are used as models of formulas rather than more complex Kripke style structures.

Let's define a stepping theory as a pair $T = (R, C_k)$, where R is a finite set of rules of the form Eq. (5), C_k is the model run clock as defined above (hereinafter, the stepping theory clock), which in step theories is always a finite set of natural numbers. For any set theory of the form $T = (R, C_k)$ let $R\ [q]$ refer to the set of all rules whose consequent is q. The set of literals forming the antecedent of rule r will be designated as $A(r)$. Let Lit_T be the set of all literals occurring in the rules of stepping theory T. *The belief set* of stepping theory $T = (R, >)$ is a set of the form $\{now(t) \cup L_{T^t}$, where t is a natural number or 0 representing a point in time on the clock C_k of this stepping theory, $L_{T^t} \subset Lit_T$. Let's consider operator ϑ_T that transforms the belief sets into other belief sets in such a way that if B is a belief set such that $now(t) \in B$, then $now(next\ t) \in \vartheta_T(B)$. The sufficient conditions under which the literals will belong to belief set $\Theta_T(B)$ will be different for different step theories, see, for example, [12] and [13]. In the next section we will look at these conditions as they apply to a system of stepping theories with two kinds of negation in the context of their argumentation semantics.

Now let B be the belief set of theory T, such that literal $now(t) \in B$. Then B is a *quasi-fixed point* of operator ϑ_T if and only if $B\ now(t) = \vartheta_T(B)\ now(next\ t)$ for any t. *The history* in stepping theory T is a finite sequence of belief sets $\boldsymbol{B} \cdot \boldsymbol{B}(i)$ is the i^{th} member in the history, $\boldsymbol{B}(0) = now(0)$, for any t $\boldsymbol{B}(rank(t) + 1) = \Theta_T(\boldsymbol{B}(rank(t)))$. The last element in the history is a belief set designated as B_{fin}, (*final*). It is the smallest quasi-fixed point of operator ϑ_T, such that if $now(n) \in B_{fin}$, then for any j, if $later(j)$ or $\neg\ later(j) \in A(r)$, where r is a rule from set R, then $n > j$. *An inference step* in stepping theory $T = (R, C_k)$ is any pair of the form $(\boldsymbol{B}(i)$, $\boldsymbol{B}(i+1))$, and *the inference step number* is the number equal to $(i+1)$. The consequent (*t-consequent*) of stepping theory T is a literal belonging to belief set B_{fin} $(\boldsymbol{B}(rank(t))$, $t \in C_k^*)$.

Stepping theories T and T' are *equivalent* if any consequent (t-consequent) of theory T is also a consequent (*t-consequent*) of theory T'.

4. Stepping Theories with Two Kinds of Negation

4.1. Stepping Theories and Self-Discovery Principle

The principles of time marking and controversy detection are fully actualized in presently known systems of stepping theories [12], [13], [14] and [15] which does not apply to actualization of the self-awareness principle. As a consequence of this fact, the language of rules for stepping theories fails to clearly express an assertion that a certain formula (a literal, in this case) is unknown by a certain point of time to the agent, whose reasoning is modeled by means of this stepping theory. However, as it was said above, such possibility is quite sought for in regulating a cognitive process in hard time-limit conditions.

An approach aimed at overcoming this shortcoming of presently existing formalisms of stepping theories, connected with introduction of a new kind of negation in the language of rules for stepping theories is given below. This kind of negation is somewhat analogous to negation as failure or is identical to *negation by default* [2] which is known to logical programming and termed as operator not.

4.2. Subjective Negation

We will call the new kind of negation subjective negation and denote it as operator not^t. While the meaning of *not* q expression is a failed effort to infer the q literal by means of a given logical program, the $not^t\ q$ expression in the antecedent of the rule of stepping theory means that the agent failed to infer the q literal by the current point in time (= on a given inference step). More formally, in case when $q \in \boldsymbol{B}(i)$, the r rule $not^t\ q \in A(r)$ on step $(\boldsymbol{B}(i), \boldsymbol{B}(i+1))$ won't work. The kind of negation, which had been in existence in the language of rules of stepping theories and had been termed by means of unary logical connective, hereafter will be called *strong negation*, i.e. the way this kind of negation is called in logical programming. In this case the same term is explained by the fact that this kind of negation has the same value both in logical programs and stepping theories and thus has the same denotation. It is interesting to note that historically the term *negation as failure* had occurred in logical programming earlier (in the early 70s of the last century) than the term *strong negation*, introduced by M. Gelfond and V. Lifschitz [16], while the term strong negation had occurred prior to the term *subjective negation* in stepping theories. Nevertheless, we will call the stepping theories with two kinds of negation the extended stepping theories, similarly to the extended logical programs (which contain two kinds of negation as well), introduced by M. Gelfond and V. Lifschitz.

4.3. Extended Stepping Theories: Syntax Changes

As opposed to stepping theories described above, the syntax of the extended stepping theories has undergone the following changes:

- The not^t subjective negation operator has been added to the alphabet of the rules of the extended stepping theories.
- The rules of the extended stepping theory look as follows:

  $$N : a_1 \wedge a_2 \ldots \wedge a_m \wedge not^t c_1 \wedge not^t c_2 \wedge not^t c_n \Rightarrow b, \tag{6}$$

  where N is a line of symbols denoting the name of the rule, b is a propositional literal, $a_1, \ldots, a_m$ are propositional literals, or the first order logic language literals of $\neg later(j)$ kind or $\neg later(j)$ kind, where j is a natural number, and $c_1, \ldots, c_n$ are propositional literals.

Hereafter any propositional literal of kind a_i which is not preceded by not^t subjective negation operator will be called an objective literal. Any literal of $not^t c_j$ kind will be hereafter called a subjective literal.

Here it should be noted that any further reasoning will remain valid even in the case when the first order logic literals without functional symbols, i.e. with the finite Herbrand universe, are used instead of the propositional literals.

4.4. Argumentation Semantics for Extended Stepping Theories

The argumentation theory [17] has proved to be quite fertile in presenting of non-monotonic reasoning [2].

It has been established that some well-known non-monotonic formalisms correspond to various specific instances of reasoning structure. It has been of much success to derive in the terms of the theory of argumentation quite elegant definitions for their consequence relation. Argumentation semantics for formalisms of active logic stepping theories was suggested in some works [13] and [14]. A variant of such kind of argumentation semantics with the consideration of specific features of extended stepping theories will be given below.

As basic elements, reasoning systems usually have a certain *logical language* and definitions for *argument*, *conflict between arguments* and *argument status*. The latter three elements are frequently used to determine the consequence relation. A reasoning system, consisting of the mentioned elements, constructed with a consideration of active logic stepping theories features, will be presented below.

Definition 1.

Let $T = (R, C_k)$ is an extended stepping theory. And let an argument for T be called:

- any literal (of the first order logic) of l kind $later(t)$, or $\neg later(t)$, where $t > 0$, for which there exists rule $r \in R$, so that $l \in A(r)$,
- sequence of rules $\mathrm{Arg} = [r_1, \ldots, r_n]$, where $r_1, \ldots, r_n \in R$, so that for any $1 \geq i \geq n$, if $p \in A(r_i)$, where p is the objective propositional literal, then there might be such $j < i$ that $r_j \in R[p]$,
- any subjective literal of $not^t\ q$ kind.

For this $T = (R, >, C_k)$ stepping theory *a set of all its arguments* will be indicated Args_T. If the first order logic literal is $later(t)$ ($\neg later(t)$) kind, then we'll call such reason limiting (the function of the other reasons after and before in time).

Argument of $\mathrm{Arg} = [r_1, \ldots, r_n]$ kind is called *supporting argument*. Propositional literal $b_{iff.}$ is called a *conclusion* of supporting argument $\mathrm{Arg} = [r_1, \ldots, r_n]$, when $r_n \in R[b]$.

Argument of not^t q kind is called a *subjective argument*.

Any subsequence of $[r_1, \ldots, r_n]$ sequence, meeting the definition 1 is called *supporting subargument* of argument $\mathrm{Arg} = [r_1, \ldots, r_n]$. Limiting, or subjective argument is a subargument of the argument $\mathrm{Arg} = [r_1, \ldots, r_n]$ if a corresponding first-order logic literal, or a subjective literal is included in the antecedent of any of the rules $r_1, \ldots, r_n$.

Any supporting subargument of the argument $\mathrm{Arg} = [r_1, \ldots, r_n]$ is called its *maximum subargument* if a literal being a conclusion of the said subargument is included in the antecedent of r_n. rule. Limiting, or subjective argument is the maximum subargument of argument $\mathrm{Arg} = [r_1, \ldots, r_n]$ if a corresponding literal is included in the antecedent of r_n rule.

Example 1. *Let the R_1 set of $T_1 = (R_1, Ck_1)$ extended stepping theory consist of the following elements:*

$N1 : \Rightarrow p,$
$N2 : p \Rightarrow q,$
$N3 : q \Rightarrow r,$
$N4 : later(4) \wedge not^t\ r \Rightarrow \neg task\_is\_solved,$
$N5 : \neg later(4) \wedge r \Rightarrow task\_is\_solved,$
$N6 : task\_is\_solved \Rightarrow task\_is\_solved.$

Supporting arguments of T_1 theory are $\mathrm{Arg}_1 = [N1, N2, N3, N5, N6]$ and all of its supporting subarguments, as well as $\mathrm{Arg}_2 = [N4]$. $\mathrm{Arg}_3 = [N1, N2, N3, N5]$ supporting subargument is the maximum supporting subargument of argument Arg_1. $\mathrm{Arg}_4 = \neg later(4)$ and $\mathrm{Arg}_5 = later(4)$ are limiting arguments of T_1 theory (and corresponding subarguments of arguments Arg_1 and Arg_2). $\mathrm{Arg}_6 = not^t\ r$ is a subjective subargument of argument Arg_2.

Going over to definition of an argument status and a conflict between arguments of extended stepping theories, we should take into consideration that unlike other systems of argumentation, where the interrelations of various arguments are studied "in statics", here, speaking informally, we have to discuss the development of these interrelations through time (temporally), i.e. in steps of inference under the extended stepping theory. On a certain step of inference, a particular argument may not have been constructed, i.e. fail to become active, while after putting into action it may be in action until the time point (step of inference) of its withdrawal from action. The latter means that the given argument (on the given step) is denied (=disposed of) due to the consequences of a conflict with other arguments. Thereby, the notion of "putting an argument into action" plays the key role in determining the status of the arguments and a conflict between them. For the purpose of simplicity hereafter we will refer to the step of inference, numbered i as to simply step i.

Definition 2.

Putting arguments of extended stepping theories in action is performed by the following rules:

- Any limiting argument of $later(t)$ kind is put into action on step i, so that $clock(i-1) < t \leq clock(i)$. Any limiting argument of $\neg later(t)$ kind is put into action on step 1 (at the time "$clock(0)$").

- If the supporting argument has no subarguments, it is put into action on step 1 (at the time "$clock(0)$").
- Any supporting argument is put into action on step i, if all of its subarguments are put into action on the previous steps and there is maximum subargument put into action on step $(i-1)$ (at $clock(i-2)$).
- Any subjective argument is put into action on step 1 (at $clock(0)$).

Definition 3.

Withdrawal of limiting arguments from action is performed by the following rule: Any limiting argument of $\neg later(t)$ kind is withdrawn from action on step $i+1$, where i is such that $clock(i-1) \leq t < clock(i)$.

All the other arguments, including limiting arguments of $later(t)$ kind, after their introduction into subsequent steps of inference, they have the status *active*.

The notion of arguments attacking each other which is present in practically all argumentation systems, has its specific features in active logics.

Definition 4.

Arg_1 supporting argument attacks the other supporting arguments with conclusion of q, or subjective argument $\mathrm{Arg}_2 = \mathrm{not}^t\ q$ on step it. and $t.t.$, when:

- conclusion of Arg_1 is literal "$-q$";
- Arg_1 and all of its subarguments are active on step i;
- none of the Arg_1 subarguments is attacked on step i by none of the other supporting arguments.

With an exception of a metaliteral of $now(i)$ kind, any belief set of the extended stepping theory consists of objective literals which are supporting arguments. The definition given below establishes necessary and sufficient condition of attributing of the objective literal to a belief set in the extended stepping theory.

Definition 5.

Let $\mathbf{B}(i)$ be a belief set of a certain extended stepping theory. Objective literal $q \in \mathbf{B}(i)$ if:

- there is a given Arg_1 extended stepping theory supporting argument whose conclusion is literal q;
- Arg_1 and all of its subarguments are active on step i and Arg_1 is put into action not later than on step $i-1$;
- None of Arg_1 subarguments is attacked on step i by no other supporting arguments.

It should be noted that the supporting argument obtains an ability to attack other arguments one step earlier than it obtains an ability to support an objective literal which is its conclusion. Thus, any contradiction in belief sets is eliminated. Some examples of the extended stepping theories will be discussed below.

4.5. Examples of Extended Stepping Theories

Let us continue review of the extended stepping theory $T_1 = (R_1, C_{k1})$. Example 1 (cont-d). Let $C_{k1} = [0,1,2,3,4,5,6]$. In this case the history of the extended stepping theory looks as follows: $\boldsymbol{B}(0) = now(0)$, $\boldsymbol{B}(1) = now(1), p$, $\boldsymbol{B}(2) = now(2), p, q$, $\boldsymbol{B}(3) = now(3), p, q, r$, $\boldsymbol{B}(4) = B_{fin} = now(4), p, q, r, task\_is\_solved$.

Note that the supporting argument $\mathrm{Arg}_2 = [N4]$ is put into action at the time point 5 on clock C_{k1}, but an objective literal $\neg task\_is\_solved$ has not appeared in the belief set $\boldsymbol{B}(6)$, as his subjective subargument $not^t\ r$ is under attack of supporting argument Arg_7 (one of subarguments of supporting argument Arg_1) at the time 2 on C_{k1} when the supporting Arg_7 is put into action. Hence, only arguments of meta-predicate $now(.)$ $(now(4) \in \boldsymbol{B}(4).\ now(5) \in \boldsymbol{B}(5), now(6) \in \boldsymbol{B}(6))/$ constitute the difference of the belief sets $\boldsymbol{B}(5)$ and $\boldsymbol{B}(6)$ from the belief sets $\boldsymbol{B}(4) = B_{fin}$.

Regarding information aspect, this example can be interpreted as follows: the task has been successfully solved, because an objective literal r was inferred in due time.

Below we show extended stepping theory $T_2 = (R_2, C_{k2})$, where $R_2 = R_1$, $C_{k2} = [0,1,2,5,6,7,8]$. History of this stepping theory is as follows: $\boldsymbol{B}(0) = now(0)$, $\boldsymbol{B}(1) = now(1), p$, $\boldsymbol{B}(2) = now(2), p, q$, $\boldsymbol{B}(3) = now(5), p, q, r$, $\boldsymbol{B}(4) = B_{fin} = now(6), p, q, r, \neg task\_is\_solved$.

In this history, though the supporting argument Arg_1 is put into action at time point 5 on C_{k2}, this argument "could not" support the objective literal $task\_is\_solved$, which is its conclusive inference, due to the fact that its limited subargument $\neg later(4)$ has ceased to be active at time 5 on C_{k2} clock. As a result, supporting argument $\mathrm{Arg}_2 = [N4]$ that was put into action at time 5 on C_{k2}, supported an objective literal $\neg task\_is\_solved$, starting from time 6 on C_{k2}. Essentially, this example corresponds to the situation when one of the steps of inference (note that actually it is a deductive cycle) turned out to proceed unexpectedly longer than the expected time. Note that it is impossible to identify such a situation in the Active Logics systems, which lack time granulation, and is difficult to identify it in stepping theories, which do not include subjective negation.

5. Conclusion

The formalism of Active Logics extended stepping theories presented in this article allows for possible use of two types of negation. In addition to *strong negation* that has been used in other formalisms of stepping theories, the subjective negation was introduced, which is directly related to actualization of the self-awareness principle, allowing the agent to recognize and express explicitly not only what he knows but also what he does not know at the given moment of time. This fact improves expressive power and, specifically, temporal sensitivity of extended stepping theories, compared to formalisms of step logics that have existed until the present day.

Due to some 'space' constraints this article does not cover matters relating to paraconsistency of argumentation semantics of extended stepping theories, but there is no doubt that the prorosition about paraconsistency of the said semantics can be proved *mutadis mutandis* similar to the method applied in [15] for argumentation semantics of stepping theories (only with strong negation).

Acknowledgment

This work was supported by the RFBR.

References

[1] BURNS, A. and A. WELLINGS. *Real-time systems and Programming Languages*. Boston: Addison-Wesley, 2001. ISBN 978-0321417459.

[2] VAGIN, V. N., Y. Y. GOLOVINA and A. A. ZAGORYANSKAYA. *Dostovernyy i pravdopodobnyy vyvod v intellektual'nykh sistemakh*. Moscow: Fizmatlit, 2008. ISBN 978-5-9221-0962-8.

[3] BEZHANISHVILI, M. N. *Logika modal'nostey znaniya i mneniya*. Moscow: KomKniga, 2007. ISBN 978-5-484-00779-0.

[4] FAGIN, R. and J. Y. HALPERN. Belief, awareness and limited reasoning. *Artificial Intelligence*. 1987, vol. 34, iss. 1, pp. 39–76. ISSN 0004-3702. DOI: 10.1016/0004-3702(87)90003-8.

[5] ELGOT-DRAPKIN, J. *Step Logic: Reasoning situated in time*. Maryland, 1988. Dissertation Thesis. University of Maryland.

[6] VINKOV, M. M. and I. B. FOMINYKH. Reasoning about knowledge and the problem of logical omniscience. Part I: The modal approach (Rassuzhdeniya o znaniyakh i problema logicheskogo vsevedeniya. Chast' I: Modal'nyy podkhod). *Iskusstvennyy intellekt i prinyatiye resheniy*. 2011, vol. 2011, iss. 1, pp. 34–47. ISSN 2071-8594.

[7] VINKOV, M. M. and I. B. FOMINYKH. Reasoning about knowledge and the problem of logical omniscience. Part II: The inclusion of temporal variables in the meta-language (Rassuzhdeniya o znaniyakh i problema logicheskogo vsevedeniya. Chast' II: Vklyucheniye temporal'nykh peremennykh v metayazyk). *Iskusstvennyy intellekt i prinyatiye resheniy*. 2011, vol. 2011, iss. 1, pp. 48–57. ISSN 2071-8594.

[8] ALECHINA, N., B. LOGAN and M. WHITSEY. A complete and decidable logic for resource-bounded agents. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*. New York: IEEE, 2004, pp. 606–613. ISBN 1-58113-864-4.

[9] HOVOLD, J. *On a semantics for active logic*. Lund, 2011. M.A. Thesis. Lund University.

[10] VINKOV, M. M. Time, as the external entity in the simulation argument rational agent with limited resources (Vremya, kak vneshnyaya sushchnost' pri modelirovanii rassuzhdeniy ratsional'nogo agenta s ogranichennymi resursami). In: *Trudy XI-y natsional'noy konferentsii po iskusstvennomu intellektu s mezhdunarodnym uchastiyem KII-2008*. Moscow: ACM, 2008, pp. 53–61. ISBN 978-5-9905686-0-0.

[11] LIU, C. and M. A. ORGUN. Verification of reactive systems using temporal logic with clocks. *Theoretical Computer Science*. 1999, vol. 220, iss. 2, pp. 377–408. ISSN 0304-3975. DOI: 10.1016/S0304-3975(99)00008-0.

[12] VINKOV, M. M. Active logic in terms of the well-founded semantics of logic programs with the priorities (Aktivnaya logika s tochki zreniya fundirovannoy semantiki logicheskikh programm s prioritetami). In: *Trudy konferentsii KII'2004*. Moscow: SBW, 2004, pp. 30–38. ISBN 978-5-9905686-0-0.

[13] VINKOV, M. M. and I. B. FOMINYKH. Argumentation semantics for theories stepper active logic granulation time (Argumentatsionnaya semantika dlya shagovykh teoriy aktivnoy logiki s granulyatsiyey vremeni). *Iskusstvennyy intellekt i prinyatiye resheniy*. 2015, vol. 2015, iss. 3, pp. 3–9. ISSN 2071-8594.

[14] VINKOV, M. M. Argumentation semantics for theories stepper active logic (Argumentatsionnaya semantika dlya shagovykh teoriy aktivnoy logiki). In: *KII-2006: sb. nauch. tr. X natsion. konf. s mezhdunar. uchast.* Moscow: ACM, 2006, pp. 64–72. ISBN 978-5-9905686-0-0.

[15] FOMINYKH, I. B. and M. M. VINKOV. Paraconsistency of Argumentation Semantics for Stepping Theories of Active Logic. In: *Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'16)*. Sochi: Springer, 2016, pp. 191–206. ISBN 978-3-319-33608-4. DOI: 10.1007/978-3-319-33609-1_15.

[16] GELFOND, M. and V. LIFSCHITZ. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*. 1991, vol. 9, iss. 3, pp. 365–385. ISSN 0288-3635. DOI: 10.1007/BF03037169.

[17] DUNG, P. M. Negation as hypothesis: on abduction foundation for logic programming. In: *Proceedings of the 8th International on Logic Programming*. Beijing: DBLP, 1991, pp. 3–17. ISBN 0-262-69177-9.

About Authors

Mikhail M. VINKOV was born in Moscow, Soviet Union. He is a Candidate of Technical Sciences, associate Professor of Bauman Moscow State Technical University (BMSTU), Faculty of Robotics and Cybernetics. He defended his Candidate of Technical Sciences (equivalent to Ph.D.) degree in 1994. He started his work as an associate professor of BMSTU in 2001. Mikhail Vinkov is a member of Russian Association for Artificial Intelligence, he has authored of more than 70 scientific works and 1 monographs. His scientific interests include knowledge-based hard real time system, reasoning situated in-time, temporal non-monotonic reasoning, inductive learning etc.

Igor B. FOMINYKH was born in Moscow, Soviet Union. He is a Doctor of Technical Sciences, a professor of Moscow Power Engineering Institute (MPEI), Department of Applied Mathematics. He defended his Candidate of Technical Sciences (equivalent to Ph.D.) degree in 1983. He started his work as an head of the laboratory of NICEVT (Research Center of Electronic Computer Engineering) in 1986. He started his work as an head of department RITAP in 1993. He defended his Doctor of Technical Sciences degree in 2000. He have been working as a professor of MPEI from 2014. Professor Fominykh is a member of Russian Association for ArtifIcial Intelligence. He has authored of more than 100 scientific works and 6 monographs. His scientific interests include applied and not classical logic, reasoning situated in-time, temporal non-monotonic reasoning, dynamic intellectual system, artificial intelligence, etc.