Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Intelligent control via new efficient logics ⋆

Stanislav N. Vassilyev * **Artem V. Davydov** **
Alexander K. Zherlov ***

* *Institute of Control Sciences, Russian Academy of Sciences, Russia, 65 Profsoyuznaya st., Moscow, 117997, Russia (e-mail: snv@ipu.ru)*
** *Institute for System Dynamics and Control Theory, Siberian Branch, Russian Academy of Sciences, 134 Lermontov St. Irkutsk 664033, Russia (e-mail: artem@icc.ru)*
*** *Institute for System Dynamics and Control Theory, Siberian Branch, Russian Academy of Sciences, 134 Lermontov St. Irkutsk 664033, Russia*

Abstract: An approach to devising the upper level of control systems for multiagent systems has been developed. Problems related to automation of action planning, including those bound up with the mode of cooperative mission performance, are considered. A new efficient logic apparatus for intelligent control is proposed.

1. INTRODUCTION

This paper discusses issues related to improving the potential of intelligent control of moving objects or groups of objects. This improvement is based on development and implementation new logic methods of knowledge representation and processing.

The term *intelligent control* is becoming standard in the field of computer-aided control (see Sinha (1996); Åström (1992)).

We have to agree with Åström (1992), even if the word *intelligence* is interpreted in a quite restrictive sense, it appears that contemporary control systems have pass a long way before these can justify the term *intelligent*.

Generally speaking, real-time knowledge-based (KB) systems have to constantly refer to what has happened, what is going on, and what may happen (see Gabbay (1992)). Furthermore, KB systems have to cope with the problem of interaction with a constantly evolving world. As far as planning is concerned, these have to be capable of predicting changes, proposing and triggering actions on the basis of these predictions. These systems have to notice, when definite predictions are no longer plausible. All this necessitates the appearance of expressive languages and powerful inference formalisms. However, usually, instead of operating with a general temporal logic, the requirement to reasoning have to be restricted (see Ramamonjison (1995)) in the aspect of both the possibility of expression of temporal relations in the form of production rules and implementation of some heuristic structurization of control processes, e.g., in the form of a multi-agent KB system (MAS) with some time- or data-driven meta-rules related to interaction of these agents.

In section 2 below a new expressive logical language $\boldsymbol{LF}$ and new method of deduction are described. The formulas of this language consist of some large-dimension structural elements such as type quantifiers. So, there are only two logic symbols $\forall$ and $\exists$ in $\boldsymbol{LF}$, and these are used as a set of connectives of $\boldsymbol{LF}$. A new logic calculus $\boldsymbol{JF}$ and complete strategies of computer-aided (automated) deduction, which are based on a unique and unary inference rule, have been developed. This calculus possesses many other properties, which provide for the reduction of the combinatorial complexity of deduction in comparison with other known systems of computer-aided deduction, e.g., resolution and Hentzen-type systems.

The principal ideas behind the proposed technique are outlined, and an example of application of MAS for the purpose of control of moving objects are considered.

2. A NEW LOGIC FOR DYNAMICAL MODELLING AND CONTROL

The proposed formalism of representation and processing knowledge is intended to further develop a logic calculus $\boldsymbol{J}$ of positive-constructed formulas (PCFs) (see Vassilyev (2000)).

2.1 A new language $\boldsymbol{LF}$ with descriptive semantics

Let us denote the set of all conjuncts as *Con* and assume that *a conjunct* be either a finite set of usual atomic formulas (atoms) of the 1-st order language or $\mathbf{F}$, where $\mathbf{F}$ satisfies the property $A \subset \mathbf{F}$ for each $A \in Con$. The empty conjunct is denoted by $\mathbf{T}$. Atoms of any conjunct (except for $\mathbf{T}$ and $\mathbf{F}$) may contain individual variables, individual constants and function letters.

The language $\boldsymbol{LF}$ of *positively constructed formulas* (PCFs) is defined as follows:

⋆ The work has been conducted with the financial support of the Program of Fundamental Research N 22 of the Presidium of Russian Academy of Sciences (under the scientific coordination of Prof. F.Chernousko) and the Program of Supporting the Leading Scientific Schools of Russia, SS-9508.2006.

978-1-1234-7890-2/08/$20.00 © 2008 IFAC 10.3182/20080706-5-KR-1001.0170

(1) if $A \in Con$, then $\mathbf{K}X{:}A$ is a **K**-formula, where X is a set of individual variables, $\mathbf{K} \in \{\forall, \exists\}$;
(2) if $B \in Con$, Φ is a set of **K**-formulas, then $\mathbf{M}Y{:}B\ \Phi$ is an **M**-formula, where $\mathbf{M} \in \{\forall, \exists\}, \mathbf{M} \neq \mathbf{K}$, Y is a set of individual variables;
(3) each PCF is either an $\forall$-formula or an $\exists$-formula.

Let nodes of the PCF's tree-structure be called the *positive quantifiers* (PQs), and without loss of generality, we usually assume that

I) a root of PCF is $\forall{:}\mathbf{T}$ (the corresponding sets of variables and atoms are empty ones),

II) each leaf is an $\exists$-node, i.e. it contains the sign $\exists$.

Any node $\exists X{:}A$ that immediately follows the root of PCF is called the *base* of the PCF (let us speak that A is a base of facts). A subformula, whose root is a base of the whole PCF, is called the *basic subformula*. Let any of the immediate successors $\forall Y : B$ of a base $\exists X : A$ be called *questions* to $\exists X{:}A$.

The semantics of PCF $\mathcal{F}$ is defined by a common semantics of a corresponding formula $(\mathcal{F})^*$ in the 1-st order predicate calculus:

(1) if $A \in Con, A \notin \{\mathbf{F}, \mathbf{T}\}$, then $A^{\&} = \&\{\alpha{:}\alpha \in A\}$, $\mathbf{F}^{\&} = False$, $\mathbf{T}^{\&} = True$ (propositional constants);
(2) let $X = \{x_1, \ldots, x_m\}$, then

$$(\exists X{:}A\ \Phi)^* = \exists x_1 \ldots \exists x_m (A^{\&} \& (\Phi)^*),$$
$$(\forall X{:}A\ \Psi)^* = \forall x_1 \ldots \forall x_m (A^{\&} \rightarrow (\Psi)^*),$$
$$\text{where } (\Phi)^* = \&\{(\alpha)^* : \alpha \in \Phi\},$$
$$(\Psi)^* = \vee\{(\alpha)^* : \alpha \in \Psi\}.$$

The common concepts of logic satisfiability, validity, inconsistency, equivalence, free and bound occurrences of variables, etc. for a PCF $\mathcal{F}$ are understood likewise for $(\mathcal{F})^*$. Suppose that

III) inside each of the basic subformulas, any variable cannot be free and bound simultaneously, furthermore, it cannot be bounded by different PQs simultaneously.

Proposition 1. The language ***LF*** of PCFs is complete w.r.t. expressibility in the 1-st order predicate calculus.

The following theorem formulates a statement on some comparative complexity of PCF-representation vs the classical one.

Theorem 2. $\forall k > 0$ there exists a sequence of the Boolean functions $f_1, \ldots, f_n$ such that

$$\mathcal{C}(n)_{PCF} \times k^{k^{\frac{n-1}{2}}} < \mathcal{C}(n)_{CNF},$$

where $\mathcal{C}(n)_{PCF}$ is the complexity of PCF-representation of the f_n, and $\mathcal{C}(n)_{CNF}$ is the complexity of representation of f_n in the conjunctive normal form.

*2.2 A new calculus **JF** with descriptive semantics*

In the process of reasoning, one often proves a statement $\mathcal{F}$ by refuting its negation. We intend to proceed similarly.

A question $\forall Y{:}B$ to a base $\exists X{:}A$ has *an answer* Θ iff Θ is a mapping (substitution) $Y \rightarrow H^\infty$ and $B\Theta \subseteq A$, where H^∞ is a Herbrand universe based on variables from X, constants and functional letters that occur in $A \cup B$. If $X = \varnothing$, and there are no constants and functional letters in $A \cup B$, then $H^\infty \stackrel{df}{=} \varnothing$.

Now we proceed to the definition of the unique inference rule. If PCF $\mathcal{F}$ has the structure $\forall{:}\mathbf{T}\{\Psi,\ \exists X{:}A\ \ \Phi\}$, where Ψ is a list of other basic subformulas (subtrees) of $\mathcal{F}$, and Φ contains a subformula $\forall Y : B\ \{\exists Z_i : C_i\ \ \Psi_i\}_{i=\overline{1,k}}$, then the result $\omega\mathcal{F}$ of application of *the unary inference rule* ω to the question $\forall Y{:}B$ with the answer $\Theta : Y \rightarrow H^\infty$ represents the formula

$$\omega\mathcal{F} = \forall{:}\ \ \mathbf{T}\ \{\Psi, \{\exists X \cup Z_i{:}A \cup C_i\Theta\ \ \Phi \cup \Psi_i\Theta\}_{i=\overline{1,k}}\}.$$

After appropriate renaming some of the bound variables inside each subformula the expression $\omega\mathcal{F}$ shall satisfy all the requirements to PCFs. We will imply such renaming each time when we apply the rule ω. The same will relate to the following simplifying substitutions:

(1) $\exists X{:}\ \mathbf{F}\ \ \ \Phi\ /\ \exists{:}\ \mathbf{F}$, i.e. $\exists X{:}\ \mathbf{F}\ \Phi$ will be replaced by $\exists{:}\ \mathbf{F}$,
(2) $\forall{:}\ \mathbf{T}\ \{\Psi, \exists{:}\ \mathbf{F}\} /\ \forall{:}\ \mathbf{T}\ \Psi$ if $\Psi \neq \emptyset$.

Any finite sequence of PCFs $\mathcal{F}, \omega\mathcal{F}, \omega^2\mathcal{F}, \ldots, \omega^n\mathcal{F}$, where $\omega^s\mathcal{F} = \omega(\omega^{s-1}\mathcal{F}), \omega^1 = \omega, \omega^n\mathcal{F} = \forall : \mathbf{T}\ \ \exists : \mathbf{F}$, is called *a deduction* of $\mathcal{F}$ in calculus $\boldsymbol{JF} =< \forall{:}\mathbf{T}\ \ \exists : \mathbf{F},\ \omega >$ (with the axiom $\forall{:}\mathbf{T}\ \ \exists : \mathbf{F}$).

Suppose that a search strategy verifies the questions in consecutive order, without omissions (while repeating the verification only when the whole cycle of questions is over), and it does not use repeated application of ω to a question with the same Θ (*QA-method*, i.e. question-answering method of computer-aided (automated) deduction).

Note, a step of search for answers Θ is used very often in the deduction procedure, so it needs special attention. As mentioned above, an answer Θ to a question $\forall Y :$ B exists if there is a matching $B\Theta \subseteq A$, where A, B are some sets of atoms, A is a base of facts. So, we have to find all such substitutions Θ. Therefore, we have to do with the classical matching problem. In order to solve it, one may use various subsumption algorithms. We employ the algorithm proposed by S.Ferilli, N.Di Mauro, T.M.A.Basile and F.Esposito (see Ferilli (2003)). It has a polynomial complexity and allows one to find all of the matching substitutions.

Theorem 3. The calculus ***JF*** is consistent and complete, i.e. for any PCF $\mathcal{F} \vdash \neg(\mathcal{F})^*$ iff $\vdash_{\boldsymbol{JF}} \mathcal{F}$.

2.3 Simple example of automated deduction

Example 1. Consider the following example: "*Any production centre is accessible for mobile robots. Some robots have no access to any offices. Therefore, none of offices is a production centre.*" The formal record of this text in terms of the 1-st order predicate calculus is $A = (A_1 \& A_2 \rightarrow B)$, where

$$A_1 = \forall x(R(x) \rightarrow \forall y(C(y) \rightarrow A(x, y))),$$
$$A_2 = \exists x(R(x)\ \&\ \forall y(O(y) \rightarrow \neg A(x, y))),$$
$$B = \forall x(C(x) \rightarrow \neg O(x)).$$

The negation of A is $A_1 \& A_2 \& \neg B$ and $A_1, A_2, \neg B$ can be represented as the PCFs

$$A_1^{\mathrm{PCF}} = \forall xy\!:\!R(x), C(y)\ \ \exists\!:\!A(x,y),$$
$$A_2^{\mathrm{PCF}} = \forall\!:\!\mathbf{T}\ \ \exists x_1\!:\!R(x_1)\ \ \forall y_1\!:\!O(y_1), A(x_1,y_1)\ \exists\!:\!\mathbf{F},$$
$$(\neg B)^{\mathrm{PCF}} = \forall\!:\!\mathbf{T}\ \ \exists x_2\!:\!O(x_2), C(x_2).$$

Therefore, $\mathcal{F} = (\neg A)^{\mathrm{PCF}} =$

$$\forall\!:\!\mathbf{T}\ \exists\!:\!\mathbf{T}\ \{\ \forall xy\!:\!R(x), C(y)\ \ \exists\!:\!A(x,y),\ \ \forall\!:\!\mathbf{T}\ \ \exists x_1\!:\!R(x_1)\ \ \forall y_1\!:\!O(y_1), A(x_1,y_1)\ \exists\!:\!\mathbf{F},\ \ \forall\!:\!\mathbf{T}\ \ \exists x_2\!:\!O(x_2), C(x_2)\ \}.$$

We can obtain an example of deduction of $\mathcal{F}$ in $\boldsymbol{JF}$:

$\mathcal{F}$;

$$\omega\mathcal{F} = \forall\!:\!\mathbf{T}\ \ \exists x_1\!:\!R(x_1)\ \ \{\ \forall xy\!:\!R(x), C(y)\ \ \exists\!:\!A(x,y),\ \ \forall y_1\!:\!O(y_1), A(x_1,y_1)\ \exists\!:\!\mathbf{F},\ \ \forall\!:\!\mathbf{T}\ \ \exists x_2\!:\!O(x_2), C(x_2)\ \}$$

$$\omega^2\mathcal{F} = \forall\!:\!\mathbf{T}\ \ \exists x_1x_2\!:\!R(x_1), O(x_2), C(x_2)\ \ \{\ \forall xy\!:\!R(x), C(y)\ \ \exists\!:\!A(x,y),\ \ \forall y_1\!:\!O(y_1), A(x_1,y_1)\ \exists\!:\!\mathbf{F}\ \}$$

$$\omega^3\mathcal{F} = \forall\!:\!\mathbf{T}\ \ \exists x_1x_2\!:\!R(x_1), O(x_2), C(x_2), A(x_1,x_2)\ \ \{\ \forall xy\!:\!R(x), C(y)\ \ \exists\!:\!A(x,y),\ \ \forall y_1\!:\!O(y_1), A(x_1,y_1)\ \exists\!:\!\mathbf{F}\ \}$$

$$\omega^4\mathcal{F} = \forall\!:\!\mathbf{T}\ \ \exists\!:\!\mathbf{F}.$$

So, we do not destroy the formula's structure likewise in Robinson (1965), and, therefore, our deduction technique is quite compatible with the heuristics (see other properties of PCFs in the section 3).

*2.4 Language **LFC** and calculus **JFC** with constructive semantics for planning and control*

It is known that a logic approach to artificial intelligence often implies the need of constructive semantics. In some logics, classical and constructive semantics simply coincide (Prolog systems). We consider a modification $\boldsymbol{JFC}$ of the calculus $\boldsymbol{JF}$, while providing the desirable constructive semantics and constructive search for the plans of actions. For example, as far as applications are concerned, coordinated actions of n agents are ensured by the logic specification of feasibility of particular actions realizable only as a result of combined (may be simultaneous) exploitation of the functional capabilities of several agents operating as a group. The searching plan is constructed automatically, while applying deduction of a specification of the goal from the specification of functional capabilities of the controlled objects considered as separate ones or those joined into a group.

Theorem 4. Let $G_1 \to G_2$ be the problem specified by the first-order formula, where G_1 is an arbitrary formula (it describes the conditions and constructive means of problem solving), and G_2 is a goal of the form $(G_2)^{\mathrm{PCF}} = \forall\bar{x} : A\ (\exists\bar{y}_1 : B_1, \ldots, \exists\bar{y}_n : B_n)$, then the $\boldsymbol{JF}$-deduction of the formula $(G_1 \,\&\, \neg G_2)^{\mathrm{PCF}}$ is constructive (e.g. transformable to an intuitionistic deduction in the known sequential calculus LJ (see Takeuti (1978))).

Hence a limitation of the application of ω only at the expense of above formulas ($\boldsymbol{LF}$-language) gives a new calculus $\boldsymbol{JFC}$ and ensures the property of constructiveness of any deduction obtained.

The basic calculus $\boldsymbol{JF}$ allows one to construct different other modifications of its semantics without changes of both the axiom and the inference rule (again only at the expense of some limitations of usage of ω).

2.5 Nonmonotonic logic calculi. An example of application

As far as control is concerned, it is usually desirable to take into account the outdating of facts with time. Therefore, it is necessary to transform our calculus $\boldsymbol{JF}$ ($\boldsymbol{JFC}$) into a nonmonotonic form. Our modification of the calculus $\boldsymbol{JFC}$ (or $\boldsymbol{JF}$) into the form of a nonmonotonic constructive (or, resp., nonmonotonic classical) logic is based on the description of the agents' actions allowed by PCFs of the form $\forall\bar{x} : A(\bar{x}) \,\&\, B^*(\bar{x})\ \ \exists\bar{y} : C(\bar{x},\bar{y})$, where the symbol $*$ belongs to a metalanguage and is used for limitation of the use of ω. If $A(\bar{x}\gamma)$, $B(\bar{x}\gamma) \subseteq D$, where γ is a substitution of variables, and D is the base of facts, then the outdating facts $B(\bar{x}\gamma)$, which are contained in D, have to be labeled by symbol $*$ and may not be used in the process of subsequent deduction. In the similar manner, by limiting the use of the rule ω, as noted above for $\boldsymbol{JFC}$, we arrive at nonmonotonic logic $\boldsymbol{LFC^*}$ (resp. $\boldsymbol{LF^*}$).

As an example, consider the following situation. Let there be two support vessels (vessel 1, and vessel 2), two AUVs, and a definite object (sample) on the sea bottom. One of the logic specifications of feasible actions can be the statement that the specimen can be lifted by an AUV alone or, at least, by two AUVs operating jointly and delivered to the nearest vessel. Another specification could state a possibility of ascertaining – which of the two vessels is the nearest from the viewpoint of delivery of the object (sample) (vessel 1 or vessel 2). The goal can be described in the form of a specification that requires delivering the object (sample) to vessel 1 or vessel 2. The logic formalization of both this situation and the goal goes beyond Horn's formalism of Prolog but remains within the frames of applicability of calculus $\boldsymbol{JFC^*}$.

Let us examine this situation, while considering it in terms of our formalism.

Now we have to formalize the following rules:

(1) AUV can be located close to the object (sample) to catch hold of it by its manipulators.
(2) If an object and an AUV are located close to each other, and this AUV does not perform other job assignments (tasks), then either this AUV can catch hold of the object and transfer into a busy state or, if there is another AUV, then the two AUVs can both catch hold of the object simultaneously and transfer themselves into the busy state.
(3) If an AUV with the object captured are located near the support vessel, then this AUV unloads the object and returns into the free state.
(4) If two AUVs with the object are located near the support vessel, then these unload the object and return into the free state.
(5) The object can be unloaded from an AUV near the support vessel.
(6) Goal: the object is located on either the support vessel 1 or support vessel 2.

In terms of the ***LF*** language the problem is formalized as follows. First of all, consider the predicates and their interpretation.

- $R(x)$ — "x is an AUV",
- $G(x)$ — "x is an object",
- $K(x)$ — "x is a support vessel",
- $F(x)$ — "x is in free state",
- $D(x, y)$ — "x differs from y",
- $P(x, y)$ — "x is near the y",
- $T_1(x, y)$ — "x captures y",
- $T_2(x, y, z)$ — "x and y capture z".

The initial state:

$$\exists r_1, r_2, s, A, B\colon R(r_1), R(r_2), G(s), K(A), K(B), F(r_1), F(r2), D(r_1, r_2), D(r_2, r_1) \quad (0)$$

The rules:

$$\forall x, y\colon G(x), R(y) - \exists\colon P(y, x) \quad (1)$$

$$\forall x, y\colon G(x), R(y), P(y, x), F(y) - \begin{cases} \exists\colon T_1(y, x) \\ \exists\colon \mathbf{T} - \forall z\colon R(z), D(y, z), P(z, x), F(z) - \exists\colon T_2(y, z, x) \end{cases} \quad (2)$$

$$\forall x, y, z\colon G(x), R(y), T_1^*(y, x), K(z), N(x, z) - \exists\colon L(x, z), F(y) \quad (3)$$

$$\forall x, y, z, u\colon G(x), R(y), R(z), T_2^*(y, z, x), K(u), N(x, u) - \exists\colon L(x, u), F(y), F(z) \quad (4)$$

$$\forall x\colon G(x) - \begin{cases} \exists\colon N(x, A) \\ \exists\colon N(x, B) \end{cases} \quad (5)$$

The goal (more exactly, negation of the goal):

$$\exists\colon \mathbf{T} - \begin{cases} \forall\colon L(s, A) - \exists\colon \mathbf{F} \\ \forall\colon L(s, B) - \exists\colon \mathbf{F} \end{cases} \quad (6)$$

Hence to solve our problem we have to find a deduction of PCF: $\forall\colon \mathbf{T}$ (0) { (1), (2), (3), (4), (5), $\forall\colon \mathbf{T}$ (6) }.

By applying the technique proposed above we can obtain real plans, which are expected to represent the solution of the initial problem. For example, one of the possible plans based on the initial state, when the support vessel A is the nearest to the object, and there is one AUV, which can lift the object, is as follows.

(1) Basing on rule 1, the AUV is getting close to the object.
(2) Basing on rule 2, the AUV catches hold of the object.
(3) Basing on rule 5, the AUV recognizes the support vessel A as the nearest one.
(4) Basing on rule 3, the AUV is moving to the support vessel A with the object.
(5) Basing on the goal rule 6,the AUV unloads the object onto the support vessel A.

It can easily be seen, there are six possible plans, which can be generated with the aid of this formalization. Furthermore, there can be another, ever more complicated situations, which can be handled with the application of our techniques.

3. DISCUSSION

(1) Consider abovementioned peculiarities of the language ***LF***. Unlike that in predicate calculus language syntax, PCFs have rather unconventional and exquisite form.
   (a) Any formula, which is written in ***LF***, is characterized by a *large-block structure* and has *only positive quantifiers.*
   (b) Any PCF has a *simple* and *regular structure*,i.e. the formula is to some extent characterized by predictability of its structure determined by the order of $\exists$- and $\forall$-nodes, which alternate at each other branch.
   (c) The *negation* of PCF is merely obtained *by replacement* of the symbols $\exists, \forall$ and vice versa (after what canonization follows).
   (d) The PCF-representation is *more compact* than the representation in terms of the known language of clauses (see Davis (1960)) in the resolution method (see Robinson (1965)). Furthermore, it is more compact than representations in terms of standard disjunctive and conjunctive normal forms.
   (e) It is not necessary to preprocess (scolemize) the formulas by eliminating of all existential quantifiers. The scolemization procedure related to this elimination leads to elevating the complexity of terms (see Davis (1960)).
   (f) The natural structure of the knowledge in ***LF*** is *retained* better. We mean here also that the description of the knowledge in the original form does not employ "theoretical" quantifiers $\forall x$, $\exists x$. Instead of these, type quantifiers $\forall x(A \to \sqcup)$, $\exists x(A \& \sqcup)$ with simple and natural forms of conditions A (e.g., without symbols $\forall, \exists$ inside of A) are employed. The PCF structure retains mainly the original structure of knowledge, when the knowledge is written in the form of an expression constructed of atoms (and/or their negations) with the aid of positive quantifiers and logic connectives $\&, \vee$. In this case, the PCF structure may differ from the initial structure merely by appending auxiliary quantifiers $\mathbf{K}$: $\mathbf{T}$ (instead of some connectives $\&, \vee$ mentioned above) and/or replacing the negations of atoms $\neg A$ with the structures $\forall : A \ \exists : \mathbf{F}$.

Consider peculiarities (1d)–(1f) in greater detail. In order to illustrate these consider the following example.

Example 2. The formula

$$\forall x(A^{\&} \to (\exists y_1\ B_1^{\&} \vee \ldots \vee \exists y_k\ B_k^{\&})), \quad (1)$$

where $B_i^{\&} = C_1^i \& \ldots \& C_n^i$, $A^{\&} = A_1 \& \ldots \& A_l$, $i = \overline{1, k}$, in terms of the PCF-representation has $l + n \cdot k$ atoms. In terms of the language of clauses it'll assume the form

$$\mathop{\&}_{(i_1,\ldots,i_k)\in(\overline{1,n})^k} (\neg A_1 \vee \ldots \vee \neg A_l \vee C^1_{i_1} \vee \ldots \vee C^k_{i_k}), \quad (2)$$

i.e. contains $(l+k)n^k$ atoms (n^k clauses)!

It is also obvious that, except for the auxiliary quantifiers, the number of atoms in the PCF-representation is not larger than in any classical disjunctive (conjunctive) normal form.

It can readily be seen, the representation (2) of the initial formula (1) is not only more complicated but also substantially destroys the initial structure, although in ***LF*** the initial formula retains the initial structure

$$\forall x : A \ \{\exists y_1 : B_1, \ldots, \exists y_k : B_k\}.$$

The language of clauses has been used in the resolution method (see Russell (1995)) since its representation (2) – in comparison with formulas of the classical predicate calculus – was hogeneous. This has allowed J.Robinson to base on Herbrand's results (see Herbrand (1930)) and develop a popular method of automated deduction with one inference rule known as the resolution rule.

(2) Now we need to emphasize quite important advantages of calculus ***JF***, which are due not only to peculiar properties of our exquisite language, but also to the proper deductive power of the calculus. Let us consider these advantages.
   (a) Unlike that in many other known logic calculi, e.g., those of Hentzen type, our calculus ***JF*** has only one inference rule (ω). Availability of only one rule causes a smaller combinatorial space. However, this is not the main advantage of our calculus. The set of important merits of ***JF*** includes not only uniqueness of its inference rule but also its unary character (e.g., in comparison with the resolution method). Another advantage of ***JF*** – obvious from the view-point of decrescence of combinatorics – is that the inference rule ω is a large-block one, likewise the language ***LF*** itself. Such a rule aids to additional reduction of complexity of the combinatorial space (unlike that of the resolution rule which is also unique, but it is binary and small-block one).

   The implementation of our deduction system ***JF*** implies that the procedure of search for substitutions Θ in order to apply the inference rule ω is based on a subsumption algorithm proposed by S.Ferilli, N.Di Mauro, T.M.A.Basile and F.Esposito (see Ferilli (2003)), which is characterized by polynomial complexity. Application of the algorithm makes the combinatorial space for searching of deductions ever smaller.
   (b) The deduction (refutation) technique described has centered on application of ω to the questions only, i.e. to the successors of PCF roots. Application of ω only to questions is based on the properties (1a), (1b) and allows one *to focus* "*the attention* of the technique" on the *local* fragments of PCF, without any loss of completeness of the technique.
   (c) The deduction technique can be described in pithy terms of the question-answering procedure instead of technical terms of formal deducibility (i.e. in terms of logic connectives, atoms, etc.).
   (d) Owing to properties (1a), (1b), (1f), (2b), (2c), the deduction technique is *quite compatible with heuristics* of specific applications as well as with general heuristics of control of deduction. Owing to (2a), the deduction process consists of *large-block steps* and, so, is *well observable* and *controllable*.
   (e) The deduction technique offers *natural OR-parallelism*, because the refutations of basic subformulas are executed independently of one another.
   (f) The deductions obtained are quite *interpretable by man* owing to properties (2c), (2d). This interpretability is quite important from the viewpoint of man-machine applications.

   So, conceptually, language ***LF*** and calculus ***JF*** are not only machine-oriented, but also human-oriented: the implementation of the our techniques for the purpose of specific applications may use these two capabilities to some greater or lesser extent.
   (g) Due to the peculiar properties of language ***LF*** and calculus ***JF*** discussed in the section 2, our logic has another very important merit: its semantics can be modified without any changes of axiom $\exists : \mathbf{F}$ and inference rule ω. Such modifications are implemented merely by some restrictions of applying ω and allow us to transform classical semantics of calculus ***JF*** in non-monotonous semantics, constructive (intuitionistic) semantics, etc.

The calculus ***JF*** has been implemented in the form of the software system "PCF Prover", which proves to be more efficient in the sense reducing the length of deduction, e.g., in comparison with such advanced systems of computer-aided deduction as "OTTER" (see Kalman (2001)) and "Vampire" (see Riazanov (2003)).

4. CONCLUSION

Some problems related to automation of control processes have been investigated and discussed. Such automation is essential for information-based control systems of moving objects. The idea is to enhance the control potential of such systems through the use of a new method of knowledge representation and processing.

We have proposed new logic aids and considered fundamental issues related to application of these aids for the purpose of modeling functions and control of moving objects (including applications to AUV). On the basis of computer experiments with the system for constructing logic deductions we concluded that logic aids we elaborated are more advanced than known prototypes.

In the nearest future we intend to develop :

(1) a special logic programming language on the basis of PCFs;

(2) new algorithms for multisequencing of deductions in our PCFs calculus, which aid to ever more efficient implementation of the calculus on such computer clusters as MVS 1000/X;
(3) modifications of our calculi for constraint satisfaction problems and constraint logic programming with applications to intelligent control.

REFERENCES

K. J. Åström and T. J. McAvoy. Intelligent Control: an Overview and Evaluation. In: D. White, D. Sofge, editors, *Handbook on Intelligent Control*, pages 3–34, Van Nostrand Reinhold, New York, 1992.

M. Davis, H. Putnam. A Computing Procedure for Quantification Theory. *Journal of ACM*, pages 201–215, July 1960.

S. Ferilli, N. Di Mauro, T.M.A. Basile and F. Esposito. A Complete Subsumption Algorithm. In: A. Cappelli and F. Turini, editors, *AI*IA 2003:Advances in Artificial Intelligence, Proceedings of 6th Congress of the Italian Association for Artificial Intelligence*, pages 1–13, 2003.

D. Gabbay and M. Reynolds. Towards a Computational Treatment of Time. In: D.M. Gabbay, C.J. Hogger, J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 351–437, Clarendon Press, Oxford, 1995.

J. Herbrand. Recherches sur la théorie de la demonstration. *Travaux de la Soc. des Sci. et des Lettres de Varsovie*, vol. III, pages 33–160, 1930.

J. A. Kalman. *Automated reasoning with Otter.* Rinton Press, Princeton, N.J., 2001.

D. Ramamonjison, P. Morizet-Mathoudeaux and N. Lefort. A Real-Time Knowledge-Based System for Plowing Support. *11-th Conf. on AI for Applications*, pages 313–319, 1995.

A. Riazanov and A. Voronkov. Limited Resource Strategy in Resolution Theorem Proving. *Journal of Symbolic Computation*, pages 101–115, July-August 2003.

J.A. Robinson. A Machine-Oriented Logic Based on Resolution Principle. *Journal of ACM*, pages 23–41, January 1965.

S. J. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Inc., 1995.

N. K. Sinha and M. M. Gupta. Toward Intelligent Machines. In: M. Gupta, N. Sinha, editors, *Intelligent Control Systems. Theory and Applications*, pages 804–807. IEEE Press, New York, 1996.

G. Takeuti. *The Theory of Proofs.* Mir, Moscow, 1978.

S. N. Vassilyev, A. K. Zherlov, E. A. Fedosov, B. E. Fedunov. *Intelligent Control of Dynamical Systems.* FIZMATLIT, Moscow, 2000.