15th Triennial World Congress, Barcelona, Spain

LOGICAL APPROACH TO CONTROL: MATHEMATICAL BASIS AND APPLICATIONS

S.Vassilyev

Institute of System Dynamics and Control Theory of Russian Academy of Sciences, Irkutsk

Abstract: This paper presents a survey of some results which have been obtained in ISDCT in the field of development of some new methods for logical control of dynamical systems. Some fundamental difficulties of deduction problem were a barrier to the wide development and application of automated deduction in the loop of control. The main point of this paper is to show that automatic theorem proving technique can and should be used in intelligent control of complex systems. There are two basic reasons for that. The 1st reason is that the modern intelligent control systems lack the required intelligence yet. The 2nd one is determined by the merits of new logical tools which allow to overcome the obstacles of extensive application of the 1st and higher order logics in specific classes of on-line problems like control. Such logical instrument is described and discussed here with applications to moving objects.

Keywords: intelligent control, real-time, planning, logic applications, calculus.

1. INTRODUCTION

The common fundamental goals of intelligent control are: to fully utilize available knowledge of a controlled object, to control in an "intelligent manner", to improve the capability of controlling the object over time through accumulation of experiential knowledge, etc. (Vassilyev, 1997).

Among the tools of intelligent control which have gained rather high recognition in control community are a) knowledge-based control in the form of fuzzy logic regulators (and other rule-based control systems), b) neural networks, and c) genetic algorithms.

These and some other tools of artificial intelligence suitable for intelligent control were considered during last 40 years . However a fundamental problem associated with this technological development is that these intelligent control systems lack the required intelligence at present yet. We consider the required intelligence organization as a hierarchical control system which couples 1) a reflex behavior ("thoughtless" reaction), 2) rule-drived behavior and 3) general reasoning.

The 1st level is based on artificial neural networks with fast computations. The 2nd level realizes the "if-then" reasoning ("if-then" control synthesis). The 3rd level as the highest one has to use some powerful logics (deduction, induction, abduction, etc.).

The modern systems of intelligent control lack the required intelligence because they do not have the powerful 3rd level. E.g., today the most powerful tools of automated deduction are automatic theorem proving (ATP) techniques in mathematical

[1] The research was supported by the RFBR (grant N 0001-00922), FPS LSS (grant N 00-15-96037) and FGP "Integration" (grant N 2.1-186).

domains (Vassilyev *et al.*, 2000; Walther, 1984), and in contrast to that, real-world problems of control have not experienced a similar success of using such well developed automatic reasoning capabilities. The existing knowledge based systems of intelligent control, especially very popular rule-based systems, are created mostly for rather restricted classes of real-world problems. We cannot say that ATP holds much favour in real–time applications. Consider some of the reasons of such situation.

There is a contradiction between expressiveness of traditional formal languages and decidability of deduction problem. On the one hand the expressive power of propositional or some logically equivalent languages, rather regular for automatic control community, is not sufficient to create intelligent control systems which can qualify this name to great advantage. The classical propositional (nonpredicate) logical language has low expressiveness. On the other hand we need not only to express problems in higher level languages, but also effectively reason within them. The propositional logical theory is theoretically decidable. However, the theorem proving in predicate logics is more complex. The 1st and higher order logics have essentially more expressive languages, but are only semidecidable: there exists an algorithm which proves all theorems, but for any such algorithm there exist some formulas which are not theorems and cannot be recognized by the algorithm as unprovable ones. Besides it is not possible to estimate uniformly by finite number of steps the length of derivation (refutation) of theorems (their negations). Moreover, even theoretically decidable fragments of theories can be practically undecidable due to complexity of computations.

That is why, it is important to provide a practical decidability (Glushkov, 1979) when we try to build a logical machine which is not general problem solver (Newell and Simon, 1961), but has a "creative" power comparable with the power of human intelligence in special fields of activity.

In particular, the logical instrument has to preserve the global heuristic structure of first–order knowledge and to be of higher compatibility with heuristics. It seems that any progress along this line is very important. Many attempts can be found in literature. Moreover, the above mentioned observation about increasing of complexity of deduction problem with the extension of language is not absolute. To illustrate this point consider the situation with rule-based systems.

From the logical point of view the formalism of rule-based systems, e.g. production systems, PROLOG-based systems (Colmerauer *et al.*, 1973), can be often considered (when a premise and a conclusion of any rule have descriptive semantics) either as propositional formulas $\mathop{\&}_{i=1}^{n} p_i \to q$ or more generally as the Horn formulas $\forall x_1 \ldots \forall x_m \ (\mathop{\&}_{i=1}^{n} P_i \to Q)$, where $P_i,\ Q$ are atoms of 1st order language. From the practical point of view the expressiveness of such formulas is rather restricted. E.g., if it is necessary to express the disjointness of a set of consequences succeeded from a premise, then even in the Horn formalism this structure of knowledge has to be destroyed, and if the original structure a priori given has not only pure logical sense, but also heuristic meaning useful in some framework of logical derivation, then after that transforming the structure its original heuristic force will not remain valid.

When the right side of production rule has an imperative semantics (e.g., contains an instruction to change the set of rules itself), it is very hard to guarantee the soundness of the whole set of rules in complex control problems. It should be noted also that the restrictions of expressiveness of the Horn language lead to a high responsibility of designers and knowledge engineers: they have to provide a sufficient completeness of a final set of rules formalized and a sufficient efficiency of application (firing) of the rules to bring a state of a controlled object (a plant) nearer to a control goal. This work of man of experience resembles the hard work in computer programming (especially, when there are some rules with conclusions which in reflecsive manner require to change some other rules). To decrease the intense work, the extension of language is useful to remain a heuristic structure of original knowledge and to represent different knowledge in intelligent and free manner. Simultaneously this leads to extension of classes of representable and solvable control problems including not only regular automatic control problems, but also problems of automated fault identification, diagnosis, structural reconfiguration, action planning, etc. including abnormal modes of operating.

We will consider here the using automated deduction (the 3rd level) only. The problem under study is how to provide the compatibility of a logic with heuristics. The subject of this paper is to describe the possibilities to resolve the usually conflicting requirements of increasing the expessiveness of traditional logical languages and efficiency of processing the knowledge represented. We describe nontraditional logical tools and possibilities of pure logical control of dynamical systems (of course, with incorporated procedure of dealing with "nonlogical" predicates, if it is necessary). Transition to the more expressive languages and more powerful automated deduction technique allows us to expand the class of solvable control problems and to improve, in particular, the performance criteria of control system. The approach we offer allows us, at least for important wide classes of tasks, to overcome the scourge of nonmonotonicity of practical reasoning and recursiveness, and at the same time to remain in some "neoclassical" framework.

2. KNOWLEDGE REPRESENTATION

The syntax of our basic language is defined by the following way (Vassilyev and Zherlov, 1995). We use: *variables:* $x, x_1, x_2, \dots, y, y_1, y_2, \dots$; *predicate symbols:* $P, Q, \dots$; *atoms:* $P(x_{i_1}, \dots, x_{i_n}), \dots$; the set *Con* of *conjuncts* which are finite sets of atoms or **T** (true) or **F** (false), where by definition a) **T** is the empty set of atoms, and b) any conjunct A is the subset of **F** ($A \subseteq \mathbf{F}$, i.e. **F** is infinite set of atoms).

We introduce the expressions which are said to be *positive quantifiers*: $\forall X : A \overset{df}{=} \forall X(A \to \sqcup)$, $\exists X : A \overset{df}{=} \exists X(A \& \sqcup)$, where $A \in Con$, and X is a set of variables (may be, empty). The positive quantifiers $\rightthreetimes : \mathbf{T}$, $\rightthreetimes : \mathbf{F}$, $\rightthreetimes \in \{\forall, \exists\}$, are lumped together as *auxiliary* quantifiers.

Positively constructed formulas (PCFs) are defined as follows:

i) if $A \in Con$, X is a set of variables, then $(\forall X : A)$ is $\forall$-formula, and $(\exists X : A)$ is $\exists$-formula;

ii) let $B \in Con$, Y be a set of variables; if $\mathcal{F}_1, \dots, \mathcal{F}_n$ are $\exists$-formulas, then $(\forall Y : B)\{\mathcal{F}_1, \dots, \mathcal{F}_n\}$ is $\forall$-formula, and if $\mathcal{F}_1, \dots, \mathcal{F}_n$ are $\forall$-formulas, then $(\exists Y : B)\{\mathcal{F}_1, \dots, \mathcal{F}_n\}$ is $\exists$-formula;

iii) there are no other $\forall$- and $\exists$-formulas; any PCF is either $\forall$- or $\exists$-formula.

A *semantics* of PCF $\mathcal{F}$ is defined by a common semantics of a corresponding formula $(\mathcal{F}^*)$ in the 1st order predicate calculus (Vassilyev and Zherlov, 1995). The negation of PCF is obtained merely by inverting all symbols $\forall, \exists$ only. We use instead of proving PCF G the refuting its negation $\mathcal{F} = (\neg(G)^*)^L$, where $\neg(G)^*$ is the negation of the image $(G)^*$ of the PCF G in the predicate calculus, and $(\cdot)^L$ means a result of PCF-representation of the 1st order formula $(\cdot)$ in L.

Thus, we can consider any PCF as a tree structure (a graph), where a branching in $\forall$-node ($\exists$-node) means disjunction (conjunction). Without loss of generality we will assume that a) any PCF $\mathcal{F}$ is a finite set of trees, the roots and leaves of those are existential (positive) quantifiers, b) in any root with empty conjunct the set of variables is nonempty.

The formulas corresponding to these trees of the $\mathcal{F}$ are called as *basic subformulas* of $\mathcal{F}$. If the set of trees for $\mathcal{F}$ is not a singleton, then image $(\mathcal{F})^*$ of $\mathcal{F}$ is the disjunction of images of formulas, corresponding those trees, otherwise the image $(\mathcal{F})^*$ is described as above in the definition of semantics. This representation will be denoted the *canonical* form. Each PCF is taken to be in that form. Instead of $>|X : A \quad \Phi$, where $\Phi = \{\mathcal{F}_1, \dots, \mathcal{F}_n\}$, $n \geq 0$, we write also $>|X : A \; \{\Phi$. In accordance to that the formula

$$\mathcal{F} = \{\exists X : A \;\; \Phi, \Psi\}, \tag{1}$$

where $\Psi \neq 0$ means the formula $(\exists X : A \;\; \Phi)^* \vee (\Psi)^*$ and the 2nd bracket in (1) means conjunctive branching.

The root $\exists X : A$ of any tree from the set of trees of $\mathcal{F}$ is referred to as the *base* of the tree. It includes the conjunct A which is spoken of as *data base*. Any immediate successor $\forall Y : B$ of the root is designated the *inquiry* to the data base A (or the *question* to the base $\exists X : A$). Any substitution $\Theta : Y \to X$ such that the result of simultaneous substitutions of all variables y from Y by variables $x = \Theta(y) \in X$ in any atom from B belongs to A, i.e. $B\Theta \subseteq A$, is said to be the *answer* for the question $\forall Y : B$ to the base $\exists X : A$.

In order to refute the PCF $\mathcal{F}$ it is sufficient to obtain in any of its bases the atom **F** by the above mentioned *question-answering procedure* of supplementing the data bases by new atoms. The rigorous definition of this procedure is determined by an inference rule ω defined below.

Let in (1) (with arbitrary subformulas Ψ, Φ) Φ includes a subformula $\forall Y : B\{\exists Z_i : C_i \quad \Phi_i\}_{\overline{1,n}}$. Then the result $\omega\mathcal{F}$ of application of the *inference rule* ω to the question $\forall Y : B$ with the answer $\Theta : X \to Y$ is the formula $\omega\mathcal{F} = \{\{\exists X \cup Z_i : A \cup C_i\Theta \;\; \{\Phi, \Phi_i\Theta\}\}_{i=\overline{1,k}}, \;\; \Psi\}$. Any finite sequence of PCFs $\mathcal{F}, \omega\mathcal{F}, \omega^2\mathcal{F}, \dots, \omega^n\mathcal{F}$, where $\omega^s\mathcal{F} = \omega(\omega^{s-1}\mathcal{F})$, $\omega^1 = \omega$, $\omega^n\mathcal{F} = \exists : \mathbf{F}$, is called *a derivation* of $\mathcal{F}$ in $J = \langle \exists : \mathbf{F}, \omega \rangle$. The calculus J has one unary inference rule ω and one axiom scheme being contradiction. The calculus J has the soundness property: if $\underset{J}{\vdash} \mathcal{F}$, then $\vdash \neg(\mathcal{F})^*$.

Some examples of problems which have been complicated for many provers known in the literature, has been solved by our software program system QUANT/1 (Cherkashin, 1999; Vassilyev *et al.*, 2000) which implements the calculus J. Among them there is the known Shubert's steamroller problem (Walther, 1984). It should be noted that in the clause language paradigm only by development a many-sorted version of resolution this problem has been solved (Walther, 1984).

Theorem 1. (Vassilyev *et al.*, 2000). The calculus J is complete, i.e. for any PCF $\mathcal{F} \vdash \neg(\mathcal{F})^* \Rightarrow \underset{J}{\vdash} \mathcal{F}$.

Example 2. Let us consider the problem of action planning for a mobile robot when the goal is to grab some object which is located in a certain place and after that to release it in a container. The world is changeable and accordingly to the known robot's actions is absolutely predictable. Let it be the discrete time scale with instants $t_0, t_1, \ldots$ The formula for the generating the instants has the form $\forall t : T(t)\ \exists t' : T(t'), N(t,t')$, where $T(t)$ iff "t is an instant", $N(t,t')$ iff "t' is immediate successor of t".

The peculiarities of this style of formalization are as follows. Each action is described by a PCF

$$\forall \overline{x} t t' : A(\overline{x}, t), N(t,t')\ \exists : A'(\overline{x}, t'), \qquad (2)$$

where $A(\overline{x}, t),\ A'(\overline{x}, t')$ are the complete descriptions of the world in the instant t and the immediately next instant t'. If some precondition $A(\overline{x}, t)$ is the same for some different actions, then for the corresponding time interval more than one action can be planned. If the actions are inconsistent in one instant, then we need to supplement the logical derivation by the heuristics which forbids the derivation of more than one action for one time interval.

The merit of this style of formalization is its pure logical modeling the time scale and dynamics of the world (see also (Gabbay and Reynolds, 1995)). The known system STRIPS (Fikes and Nilsson, 1993) couples logic with algorithmic deletion of obsolete facts, resulting in a failure to some extent of the main goal of artificial intelligence and logic programming (to reach pure descriptive style of programming).

Yet another style of formalization which does not use the extra-variable of time and fills an intermediate place between our and STRIPS's approaches is described in (Vassilyev *et al.*, 2000).

Example 3. Consider the world without any reliable prediction. We model the alternative futures assuming that the future has not happened yet. We use two time scale of reasoning: real-time scale and abstract (more fast) time scale.

We model the properties of multiple futures in abstract time scale under actions of different admissible controls. This modeling allows to predict to some extent the evolving world (its reaction on the controls), and to estimate logically as well as to select logically the "best" control. In the real-time scale a fixed length of time interval of one step of the decision-making is limited by a priori time sampling in control system. In the abstract time scale the total duration of fast multi-step predicting immediate futures with selecting the most preferable one cannot exceed the length of the aforementioned decision-making interval of real-time scale and depends on throughput of computer. Instead of proving a given theorem we derive a priori unknown theorems as some immediate logical consequences of the past and present state of controlled object under alternative controls. By the initial instant of next interval of real-time scale some information on real world changes is entered into the control system by special sensors and is accounted as updated set of facts (atoms).

This approach has been proposed in (Vassilyev and Zherlov, 1998) and is discussed there with application to control of group of passenger elevator cabins when there are floor calls as external perturbations.

The well-known Call Assignment Method of numerical multicriteria optimization widely implemented in practice of elevators' control can be successfully replaced by our approach which is more flexible for accounting many peculiarities of specific maintenance of buildings.

3. ON CONSTRUCTIVE SEMANTICS

In many cases the classical derivation is not suitable for solving control problems. E.g., classical derivation of the formula $A \vee \neg A$, where A means "control u_0 is optimal", is trivial and gives nothing for the question on optimality of u_0; the derivation has to be constructive, when it is possible to extract from the derivation the constructive procedure of answering what is valid specifically: either A or $\neg A$?

Theorem 4. The constructive task $F \Rightarrow G$, where G has the form $\forall \bar{x}(A \rightarrow \vee \{\exists \overline{y}_i : B_i, i = \overline{1,n}\})$, may be replaced by solving the classical task $\exists : \mathbf{T}$ $\{\forall : \mathbf{T}\ F,\ \overline{G}\} \Rightarrow \exists : \mathbf{F}$ in the calculus J, i.e. by the J-refutation of the formula $\exists : \mathbf{T}\{\forall : \mathbf{T}\ F,\ \overline{G}\}$.

Example 5. Consider the simple (propositional) example of formalization of a task of structural reconfiguration of attitude control system after detecting the failure, say, of yaw angle sensor (f_3). Let the other 2 angle sensors (f_1, f_2) and 3 angle velocity sensors (g_1, g_2, g_3) are trouble free, and there are a digital P-controller (h) and on-board algorithm (H) of computing the yaw angle $\tilde{\theta}$ on the basis of 2 others and 3 angle velocities. This example presents the constructive task $F \Rightarrow G$, where $G = \forall \varphi, \psi, \theta, \varphi', \psi', \theta'\ \ \exists u$, and

$$\begin{aligned} F = \exists : \mathbf{T}\ \{ & \forall \tilde{\varphi}, \tilde{\psi}, \tilde{\theta}\ \exists u, && (h) \\ & \forall \varphi\ \exists \tilde{\varphi},\ \forall \psi\ \exists \tilde{\psi}, && (f_1, f_2) \\ & \forall \varphi' \exists \tilde{\varphi}',\ \forall \psi' \exists \tilde{\psi}',\ \forall \theta' \exists \tilde{\theta}', && (g_1, g_2, g_3) \\ & \forall \tilde{\psi}, \tilde{\theta}, \tilde{\varphi}', \tilde{\psi}', \tilde{\theta}'\ \ \exists \tilde{\theta}\}. && (H) \end{aligned}$$

The desirable structure extracted from the derivation has the form of the composition $h(f_1, f_2, H(f_1, f_2, g_1, g_2, g_3))$ (instead of $h(f_1, f_2, f_3)$ for the normal mode of operation) and eliminates the uncertainty of yaw angle θ.

Theorem 4 extends the known (Horn) part of 1st order logic which has the procedural semantics (Kowalski, 1974). A 1st order example of applying the constructive part of the calculus J in on-board telescope guidance system with automated diagnostics and reconfiguration of measurement system and with sliding mode of control synthesis has been described in (Vassilyev and Cherkashin, 1998).

4. ADVANTAGES OF PCF-FORMALISM

Consider the peculiarities of the language L. Although the elements of PCFs belong mostly to the classical predicate calculus syntax, PCFs have as a whole rather unconventional and ingenious form.

1. Any formula of L has *large-block structure* and *positive quantifiers only.*

2. Any PCF has *simple* and *regular structure*, i.e. the formula has to some degree a predictability of the structure, determined by the order of $\exists$- and $\forall$-nodes which alternate in each branch.

3. The *negation* of PCF is obtained *merely by inverting* the symbols $\exists, \forall$ (followed by canonization).

4. The PCF-representation is *more compact* than the representation in the clause language (Davis and Putnam, 1960) and more compact than representations in standard disjunctive or conjunctive normal forms.

5. It is not necessary to preprocess the formulas by the elimination of all existential quantifiers. Known Scolemization procedure for this elimination leads to increasing the complexity of terms (Davis and Putnam, 1960).

6. With the L, the natural structure of the knowledge is *preserved* better.

Let us consider the features 5, 6 in some detail. To illustrate them consider the following example.

Example 6. The formula

$$\forall x (A^{\&} \rightarrow (\exists y_1\ B_1^{\&} \vee \ldots \vee \exists y_k\ B_k^{\&})), \quad (3)$$

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

$$\&_{(i_1,\ldots,i_k)\in(\overline{1,n})^k} (\neg A_1 \vee \ldots \vee \neg A_l \vee C_{i_1}^1 \vee \ldots \vee C_{i_k}^k), \quad (4)$$

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

It is obvious also that not counting the auxiliary quantifiers, the number of atoms in PCF-representation is no more than in any classical disjunctive (conjunctive) normal form. Moreover, the following theorem is valid.

Theorem 7. (Zherlov, 1997). For all $k > 0$ there exists a sequence $f_1, \ldots, f_n, \ldots$ of Boolean functions such that the complexity of PCF-representation of f_n is $k^{k^{n-1/2}}$ times less than the complexity of representation of f_n in disjunctive (conjunctive) normal form.

It is apparent that the representation (4) of (3) not only more complicated, but also destroys substantially the original structure, although in L it preserves the original structure: $\forall x : A\ \ \{\exists y_1 : B_1, \ldots, \exists y_k : B_k\}$.

The clause language has been used in the resolution method due to homogeneity of representation (4) in comparison with the formulas of the classical predicate calculus. That has allowed to J.Robinson to create on the basis of Herbrand's results (Herbrand, 1930) the most popular method of automated deduction with the single, binary inference rule (resolution rule).

7. The calculus J has only *single, unary* and *large-block inference rule* (ω), leaving no room for much redundancy in a search space. Such a rule decreases the complexity of the search space to a greater extent than the resolution rule which is also the single rule, but it is binary and small-sized. We was able also to develop the calculus with the single rule ω, but is has been done for the PCF-representation which is essentially more attractive due to the features 1-6 than the clause language.

8. The deduction (refutation) technique described has centered on application of ω to the questions only, i.e. to the successors of the PCF roots. This is based on the features 1, 2 and allows *to focus* "*the attention* of the technique" on the *local* fragments of PCF without loss of completeness of the technique and avoiding stupid processing many irrelevent parts of the formula under refutation.

9. The deduction technique can be described in meaningful terms of question-answering procedure instead of technical terms of formal deducibility (i.e. in terms of logical connectives, atoms, etc.). This technique is easy to combine with procedures like solving 1st order logical equations for operating under *incomplete* information (Vassilyev, 1997).

10. Owing to the features 1, 2, 6, 8-9 the deduction technique is *well compatible with heuristics* of specific applications as well as with general heuristics of control of derivation. Owing to the feature 7 the derivation process consists of *large-block steps* and is *well observable* and *controllable*. Thus, all these features allow to incorporate domain specific knowledge and heuristic guidance.

11. The deduction technique offers *natural OR-parallelism*, because the refutations of basic subformulas are performed independently of one another.

12. The derivations obtained are well *interpretable by human* through the features 9, 10. This is important in man-machine applications. Thus, conceptually, the language L and the calculus J are not only machine-oriented, but also human-oriented: to a greater or lesser degree an implementation for specific application can use these both possibilities.

13. Due to the peculiarities of the language L and the calculus J there is very important merit of the logic: its semantics can be modified without any changes of the axiom $\exists : \mathbf{F}$ and the inference rule ω. Such modifications are realized merely by some restrictions of applying the ω and allow us to transform the classical semantics of the calculus J in non-monotonous semantics, constructive (intuitionistic) semantics (see the section 3), etc. A theoretical basis of such modifications with applications to control of dynamical systems is given in (Vassilyev *et al.*, 2000).

5. CONCLUSIONS

In this paper we have considered the new logical instrument with applications in intelligent control problems. We have described the logical language and calculus which have many important merits in comparison with traditional logical basic systems for knowledge representation and processing. We have described some examples of using the language and calculus which confirm the efficiency of the presented logical means.

6. ACKNOWLEDGMENT

The author would like to thank his colleagues Dr. A.K. Zherlov and Dr. E.A. Cherkashin, in co-authorship with them some results used in this survey have been obtained.

REFERENCES

Cherkashin, E.A. (1999). *Software System QUANT/1 for Automatic Theorem Proving. Cand. of Sci. Thesis.* ISDCT SB RAS. Irkutsk.

Colmerauer, A., H. Kanoui, R. Pasero and P. Roussel (1973). *Un Systéme de Communication Homme-Machine en Francais.* Rapport, Groupe d'Intelligence Artificielle.

Davis, M. and H. Putnam (1960). A computing procedure for quantification theory. *J. ACM* **7**, 201–215.

Fikes, R.E. and N.J. Nilsson (1993). STRIPS, a retrospective. *Artificial Intelligence* **59**(1-2), 227–232.

Gabbay, D.M. and M. Reynolds (1995). Towards a computational treatment of time. In: *Handbook of Logic in Artificial Intelligence and Logic Programming, vol.4, Epistemic and Temporal Reasoning.* Clarendon Press.

Glushkov, V.M. (1979). *Computer is Proving.* Vol. 2. ZNANIE. Moscow.

Herbrand, J. (1930). Recherches sur la théorie de la demonstration. In: *Travaux de la Soc. des Sci. et des Lettres de Varsovie, III.*

Kowalski, R. (1974). Predicate logic as a programming language. In: *Proc. IFIP-74.*

Newell, A. and H.A. Simon (1961). GPC, a programm that simulates human thought. *Lernende Automaten* pp. 109–124.

Vassilyev, S.N. (1997). Method of synthesis of derivability conditions for Horn and some other formulas. *Siberian Math. Journal* **38**(5), 896–906.

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

Vassilyev, S.N. and A.K. Zherlov (1995). On calculus of formulas with type-quantifiers. *Dokl. Akad. Nauk* **343**(5), 583–585.

Vassilyev, S.N. and A.K. Zherlov (1998). Logical approach to intelligent control. In: *Proc. of CESA'98.* Vol. 2. pp. 111–116.

Vassilyev, S.N. and E.A. Cherkashin (1998). Intelligent control of telescope. *Siberian J. of Industrial Mathematics* **1**(2), 81–98.

Walther, C. (1984). A mechanical solution of schubert's steamroller by many-sorted resolution. In: *Proc. of 4th National Conference on Artificial Intelligence.* pp. 330–334.

Zherlov, A.K. (1997). On the propositional pg-formulas and the realization of Boolean functions. *Optimization, Control, Intelligence* **2**, 151–155.