New Logics for Intelligent Control

E. Cherkashin, A. Postoenko, S. Vassilyev, A. Zherlov
Institute of System Dynamics and Control Theory
Siberian Branch of Russian Academy of Sciences
134, Lermontov str., Irkutsk, 664033, Russia
e-mail: vas@icc.ru, eugeneai@icc.ru

Abstract

New automatic theorem proving (ATP) techniques for application in control systems and artificial intelligence is proposed. We consider new logical languages in descriptive and constructive semantics. These languages consist of 1–st order formulas with type quantifiers. We define logical calculi of classical and intuitionistic types as well as strategies of automated reasoning. Information on results of these logical tools usage in some control problem is given.

Keywords: *Automatic theorem proving, Descriptive and constructive logics, Intelligent control*

Introduction

In the field of intelligent control it is helpful to use fuzzy logic regulators and other rule–based control systems as well as neural networks, Petri nets, discrete event dynamic models and genetic algorithms. For example, fuzzy logics and neural networks have gained recognition in control community with great violence and have advanced at present dramatically (White & Sofge 1992; Sinha & Gupta 1996).

As far as the intelligence level is concerned, amongst the above–mentioned approaches to intelligent control the knowledge–based (KB) systems hold the greatest promise for supporting high–level reasoning, although other systems can also be very beneficial as supplementary to the KB systems, when they offer promise as a faster tool in real–time response. The coming years will witness strengthening the trend to integrate the both levels of intelligence into a single control system. By now, even if the word intelligence is interpreted in a very restrictive sense, it appears that current intelligent control systems have a long way to go before they can qualify this name in full measure (Åström & McAvoy 1992). This paper is concerned with the problem of increasing the intelligence level of automatic and human–machine control systems.

A subject we elaborate upon here in the progress of (Vassilyev 1990; 1996; Vassilyev & Zherlov 1995; Vassilyev 1997) is a new automatic theorem proving (ATP) technique for constructive searching for a desirable control in automatic control systems. This application has been developed here in the progress of development and application of some original descriptive and constructive logics in the field of automated and semi–automated software (Butyrin & et al. 1997; Matrosov, Sumenkov, & Vassilyev 1991) and hardware (Patrushev & Vassilyev 1989) engineering. It should be noted that amongst the reasoning methods of KB systems ATP techniques appear as rather powerful tools particularly in mathematical and programming environment. However, we cannot say that ATP holds much favour in real–time applications. 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. On the other hand we need not only to express problems in higher level languages, but also effectively reason within them. However, the theorem proving in predicate logics is more complex. The logical instrument has to preserve the global heuristical 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. (see, e.g., in (Gabbay, Hogger, & Robinson 1994) surveys by L. Wos and R. Veroff, pp. 1–40, Ch. Walter, pp. 127–228, D.M. Gabbay, pp. 350–500).

We discuss new logical languages of positively constructed formulas (PCFs) in descriptive and constructive semantics as well as new universal methods of ATP with application in telescope guidance problem.

Transition to the more expressive languages and more powerful ATP technique allows us to expand the class of solvable control problems and to improve, in particular, the quality characteristics of control systems. The well known resolution based ATP deals with normal form where the clauses are disjunctions of atoms or their negations. Our languages of PCFs as compared with the clause language of resolutional type have essentially higher level with large structural elements in size, and the corresponding calculi have inference rules which deal with the large–sized items. This leads to essential reducing a search space and usually to shortening the derivation sequences. Very important property of the deductive system is a better compatibility with heuristics.

In particular, we propose descriptive logical calculus J. Its language is complete w.r.t. expressiveness of the language of the classical predicate calculus. The calculus contains only one axiom and one inference rule and has been equipped by complete strategy of ATP. Our ATP technique has the main merits of the resolution method, and, e.g., it is well machine–oriented. Moreover, the technique preserves the global structure of first–definable knowledge (e.g., does not eliminate the quantifiers), does not generate redundant variety of terms due to scolemization absence, has high compactness of knowledge representation; it is also easy applicable for human being.

The language L of PCFs

The language L of PCFs is a first order language. The formulas of the language are represented as a tree–structures where branching nodes correspond semantically to disjunctive and conjunctive connectives with finite arity. Each node of the tree has a *type–quantifier* (TQ in the sense of N.Bourbaki). The TQs $\mathbf{K}X : A$ consist of quantifier sign $\mathbf{K} \in \{\forall, \exists\}$, a *vector of variables* X, which could be empty one, and a *type–condition* A (TC). We will restrict the range of the first–order formulas A by so–called conjuncts. The conjunct is a finite set of atoms or $\mathbf{F}$ (*"falsity"*), and besides for any conjunct A the relation $A \subset \mathbf{F}$ is hold; the empty conjunct is denoted by $\mathbf{T}$ (*"truth"*). Any conjunct $A \in Con$, where Con is a set of all conjuncts. The semantic of each branching depends on the quantifier sign in the node it origins from (see the semantics below).

The tree–structure has additional restrictions: in the leaves of the tree–structure only the existential TQs are used; the root node contains TQ $\forall\mathbf{T}$; there is the interchange of the existential and universal TQs along each path of the tree (it can be reached by inserting additional TQ–nodes where X and A are empty).

Each subformula (subtree) $\exists X\colon A\ \Psi$ which follows the root immediately is referred to as *basic subformula*, and its root node $\exists X\colon A$ as *base*. In each base $\exists X\colon \mathbf{T}$, the set X is not empty. The tree–structures we defined are referred to as *PCFs*.

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

1. if $A \in Con, A \notin \{\mathbf{F}, \mathbf{T}\}$, then $A^{\&} = \&\{\alpha : \alpha \in A\}, \mathbf{F}^{\&} = False, \mathbf{T}^{\&} = True$;
2. $(\exists X\colon A\ \Phi)^* = \exists x_1 \ldots \exists x_m (A^{\&} \& (\Phi)^*), (\forall X\colon A\ \Psi)^* = \forall x_1 \ldots \forall x_m (A^{\&} \rightarrow (\Psi)^*)$, where $\{x_1, \ldots, x_m\} = X$, $(\Phi)^* = \&\{(\alpha)^* : \alpha \in \Phi\}$, $(\Psi)^* = \vee\{(\alpha)^*\colon \alpha \in \Psi\}$.

Remark 1 In particular calculi with PCFs which will be defined below it is more convenient to consider PCFs as beginning with existential TQs (language L'), and also language L'', in which formulas the root node may be either existential or universal TQ. In these cases the given above definition of descriptive semantics is not changed, but some constructive semantics will be used too (see below Theorem 5).

The universal method of ATP

We call any immediate successors $\forall Y\colon B$ of a base $\exists X\colon A$ as *question* to $\exists X\colon A$. A question $\forall Y\colon B$ to a base $\exists X\colon A$ has *an answer* Θ iff Θ is a mapping (substitution) $Y \rightarrow TermA$ and $B\Theta \subseteq A$, where $TermA$ consists of all terms from A if $A \neq \mathbf{T}$ and $TermA = X$ otherwise.

Let PCF $\mathcal{F}$ have the structure $\mathcal{F} = \forall\mathbf{T}\ \{\Psi, \exists X\colon A\ \Phi\}$, where Ψ is a list of other PCFs as subformulas (subtrees) of $\mathcal{F}$, and Φ contains a subformula $\forall Y\colon B\ \{\exists Z_i\colon C_i\ \Psi_i\}_{i=\overline{1,k}}$ Then the result $\omega\mathcal{F}$ of application of *the inference rule* ω to the question $\forall Y\colon B$ with the answer $\Theta\colon Y \rightarrow TermA$ is the formula $\omega\mathcal{F} = \forall\mathbf{T}\ \{\Psi, \{\exists X \cup Z_i\colon A \cup C_i\Theta\ \Phi \cup \Psi_i\Theta\}_{i=\overline{1,k}}\}$ After appropriate renaming some of bound variables inside of each subformula the expression $\omega\mathcal{F}$ will satisfy all the requirements for PCFs. Such renaming we will imply always during application of ω as well as the following simplifying substitutions:
1. $\exists X\colon \mathbf{F}\ \Phi / \exists\mathbf{F}$, i.e., $\exists X\colon \mathbf{F}\ \Phi$ is replaced with $\exists\mathbf{F}$,
2. $\forall\mathbf{T}\ \{\Psi, \exists\mathbf{F}\} / \forall\mathbf{T}\ \Psi$ if $\Psi \neq \emptyset$.

Theorem 1 *For any PCF* $\mathcal{F} \vdash (\mathcal{F})^* \leftrightarrow (\omega\mathcal{F})^*$.

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 *an inference* of $\mathcal{F}$ in the calculus $J = \langle\forall\mathbf{T}\ \exists\mathbf{F},\ \omega\rangle$. The calculus J has one inference rule ω and one axiom being contradiction. According to Theorem 1 the calculus J is correct: if $\vdash_J \mathcal{F}$, then $\vdash \neg(\mathcal{F})^*$.

Theorem 2 *The calculus J is complete, i.e., for any PCF $\mathcal{F}$ if $\vdash \neg(\mathcal{F})^*$, then $\vdash_J \mathcal{F}$.*

Thus, the language of PCFs as compared with the clause language of resolutional type has essentially higher level with large structural elements in size, and the corresponding descriptive calculus have inference rule which deal with the large–sized items. The language preserves the global structure of first–definable knowledge (e.g., does not eliminate quantifiers), does not generate redundant variety of terms due to scolemization absence. In comparison with the language used in the resolution methods, compactness of representation of the knowledge can be illustrated by the following example. The formula $\forall\mathbf{T}\{\exists X_1 : A_1, \ldots, \exists X_n : A_n\}$, where each $A_i, i = \overline{1,n}$, contains m atoms, after transforming into a set of disjuncts will give m^n disjuncts.

The application of logical calculus J, with descriptive semantics, or some modification of it in solving problems with constructive semantics (action planning, computer program synthesis, automatic control, etc.) needs an additional investigation. In this case it is of purpose to make previously some conversion of J into the so–called *tasks calculus*.

The calculus J' of descriptive tasks

A J'–task is an expression $\mathcal{F} \Rightarrow \exists\mathbf{F}$ (to prove that $\mathcal{F}$ is inconsistent). In this case it is more convenient to consider that in normal form PCFs have existential TQ as a root (language L') due to a modification of the inference rule ω (see below rule RJ'). Then the J'–calculus is a calculus with one axiom:

$$AxJ': \quad \exists\mathbf{F}\{\Phi\} \Rightarrow \exists\mathbf{F}$$

and one inference rule:

$$RJ': \frac{\begin{array}{r} n \text{ tasks } \exists X \cup U_i : A \cup A_i\Theta_i\{\Phi \cup \Phi_i\Theta_i\} \Rightarrow \exists\mathbf{F}, \\ i = \overline{1,n} \end{array}}{\text{task } \exists X : A\{\Phi\} \Rightarrow \exists\mathbf{F}}$$

for some answer $\Theta : Y \to TermA$, $B\Theta \subseteq A$, where $\forall Y : B\{\exists Z_1 : A_1\{\Phi_1\}, \ldots, \exists Z_n : A_n\{\Phi_n\}\} \in \Phi$ and for any $i \in \overline{1,n}$ $\Theta_i = \Theta \cup \{z_1^i/u_1^i, \ldots, z_{k_i}^i/u_{k_i}^i\}$, $\{z_1^i, \ldots, z_{k_i}^i\} = Z_i$, u_j^i are new variables, $U_i = \{u_1^i, \ldots, u_{k_i}^i\}$. The solving the task presented in lower part of figure RJ' is reduced to n tasks from upper part of RJ'.

The Theorems 1, 2 can be adopted in *more general* case to tasks calculus J'. Let us add some notions.

We call as *a PCF–strategy* a rule ν, which determines for every J'–task, being not an example of AxJ', a set of *ν–acceptable answers*. Let admit that the PCF–strategy excludes repeating answer usage.

If ν is a PCF–strategy, then calculus $\langle J', \nu\rangle$ is defined so as new condition is added to the rule RJ', namely that the answer $B\Theta \subseteq A$ is ν–acceptable.

We call as *the strict PCF–strategy ν* a PCF–strategy that defines for each J'–task a one–element set of ν–acceptable answers. The strict PCF–strategy ν defines for each J'–task $\exists X$: A $\{\Phi\} \Rightarrow \exists\mathbf{F}$ a single search tree $\mathbf{T}_\nu(\exists X$: A $\{\Phi\} \Rightarrow \exists\mathbf{F})$, that is infinite for some tasks. The answer $B\Theta \subseteq A$ of task $\exists X$: A $\{\Phi\} \Rightarrow \exists\mathbf{F}$ which is not used *is omitted* by strict strategy ν if there exists in the tree $\mathbf{T}_\nu(\exists X$: A $\{\Phi\} \Rightarrow \exists\mathbf{F})$ an infinite branch, in which this question is not used. The strict strategy is *ν–nonomitting* if for all J'–task $\exists X$: A $\{\Phi\} \Rightarrow \exists\mathbf{F}$ there does not exist a task and an answer for this task, which is omitted by the strategy ν. Let $\bar{\mathcal{F}}$ be a representation of formula $\neg(\mathcal{F})^*$ in the language L', $\mathcal{F} \in L'$ (this representation can be done with simple inversion of quantifiers and addition of TQ $\exists\mathbf{T}$ and TQ $\exists\mathbf{F}$ as a new root and as all the new leave nodes, resp.).

Theorem 3 *If ν is a strict nonomitting strategy, then the calculus $\langle J', \nu\rangle$ is complete, i.e., for any PCFs $\mathcal{F}_1$ and $\mathcal{F}_2$ ($\mathcal{F}_1, \mathcal{F}_2 \in L'$) the following statement is hold:*

$$\mathcal{F}_1 \models \mathcal{F}_2 \iff \underset{\langle J',\nu\rangle}{\vdash} \exists\mathbf{T}\ \{\mathcal{F}_1, \bar{\mathcal{F}}_2\} \Rightarrow \exists\mathbf{F}.$$

The strict nonomitting strategy is some well organized inference search (i.e., a sequence of answers to some questions), which ensures, in principle, the deductibility of a task $\exists\mathbf{T}$ $\{\mathcal{F}_1, \bar{\mathcal{F}}_2\} \Rightarrow \exists\mathbf{F}$ in the case $\mathcal{F}_1 \models \mathcal{F}_2$. Emphasizing such notion lets us to speak about *omitting* strategies. Some omitting strategies allow to search for other, e.g., intuitionistic inferences within calculus J' and even within J (see below Theorem 6 and Corollary 1).

The calculus J_c of constructive tasks

To define a constructive (intuitionistic) analog J_c of calculus J' one must take into account the following:

- simple reducing the task $\mathcal{F} \Rightarrow \mathcal{G}$ to the task $\exists\mathbf{T}$ $\{\mathcal{F}, \bar{\mathcal{G}}\} \Rightarrow \exists\mathbf{F}$, which is admissible in classical case, is not allowed intuitionistically;
- when translating a formula of predicate calculus into some PCF some intuitionistic nuances are lost forever. So, two intuitionistically nonequivalent propositional formulas $(A \to B) \to C$ and $(A\&\neg B) \vee C$ are represented in L equally as $\forall\mathbf{T}$ $\{\exists C,\ \exists A\ \forall B\ \exists\mathbf{F}\}$. Therefore, the L is intuitionistically more poor.

Thus, by definition, the expressions $\mathcal{F} \Rightarrow \mathcal{G}$, where $\mathcal{F}$ and $\mathcal{G}$ are PCFs, are the tasks of calculus J_c (constructive analog of the calculus J), and without loss of generality one can consider that $\mathcal{F} \in L'$, $\mathcal{G} \in L''$ and if $\mathcal{G}$ is a $\forall$–formula, then it looks like $\forall\mathbf{T}$ $\mathcal{G}'$. These constructive tasks may be understood as follows: $\mathcal{F}$ is an agregation of descriptive knowledge and constructive procedures specifications (e.g., computer programs specifications), $\mathcal{G}$ is a goal specification (e.g., respectively, a specification of a computation problem, etc.).

The calculus J_c is defined as the calculus with one axiom and three inference rules ($n \in \overline{1,n}$):

$$i)\ AxC: \quad \exists\mathbf{F}\{\Phi\} \Rightarrow \mathcal{F};$$

ii) "knowledge usage":

$$R_1C_n: \frac{\begin{array}{r} n \text{ tasks } \exists X \cup U_i : A \cup A_i\Theta_i\{\Phi \cup \Phi_i\Theta_i\} \Rightarrow \mathcal{F}, \\ i = \overline{1,n} \end{array}}{\text{task } \exists X : A\{\Phi\} \Rightarrow \mathcal{F}},$$

for some answer $\Theta : Y \to TermA$, $B\Theta \subseteq A$, where $\forall Y$: $B\{\exists Z_1$: $A_1\{\Phi_1\}, \ldots, \exists Z_n : A_n\{\Phi_n\}\} \in \Phi$ and for all $i \in \overline{1,n}$ $\Theta_i = \Theta \cup \{z_1^i/u_1^i, \ldots, z_{k_i}^i/u_{k_i}^i\}$, $\{z_1^i, \ldots, z_{k_i}^i\} = Z_i$, u_j^i are new variables, $U_i = \{u_1^i, \ldots, u_{k_i}^i\}$;

iii) "$\exists$–problem solving":

$$R_2C_n: \frac{\begin{array}{r} n \text{ tasks } \exists X \cup U_i : A \cup A_i\Theta_i\{\Phi\} \Rightarrow \forall\mathbf{T}\{\Psi_i\Theta_i\}, \\ i = \overline{1,n} \end{array}}{\begin{array}{r} \text{task } \exists X : A\{\Phi\} \Rightarrow \exists Y : B\{\forall Z_1 : A_1\{\Psi_1\}, \ldots, \\ \forall Z_n : A_n\{\Psi_n\}\} \end{array}},$$

for some answer Θ : $Y \to TermA$, $B\Theta \subseteq A$,, where for any $i \in \overline{1,n}$ $\Theta_i = \Theta \cup \{z_1^i/u_1^i, \ldots, z_{k_i}^i/u_{k_i}^i\}$, $\{z_1^i, \ldots, z_{k_i}^i\} = Z_i$, u_j^i are new variables, $U_i = \{u_1^i, \ldots, u_{k_i}^i\}$;

iv) "$\vee$–problem solving":

$$R_3C_i: \frac{\exists X : A\{\Phi\} \Rightarrow \mathcal{F}_i}{\exists X : A\{\Phi\} \Rightarrow \forall\mathbf{T}\{\mathcal{F}_1, \ldots, \mathcal{F}_n\}}.$$

Indeed, for $n = 0$ the rule R_2C_n is converted in

$$R_2C_0: \frac{\exists\mathbf{F}\{\Phi\} \Rightarrow \exists\mathbf{F}}{\exists X : A\{\Phi\} \Rightarrow \exists Y : B},$$

when $B\Theta \subseteq A$ due to $\exists Y$: $B \equiv \exists Y$: B $\forall\mathbf{F}$, i.e., the upper task in R_2C_0 is a particular case of AxC.

Theorem 4 *The task $\mathcal{F} \Rightarrow \mathcal{G}$ has a solution in J_c if and only if $(\mathcal{F})^* \to (\mathcal{G})^*$ is provable intuitionistically.*

Theorem 5 *If a task $\mathcal{F} \Rightarrow \mathcal{G}$ is decidable (in J_c), then for each $\exists$–variable and $\vee$–branching of the formula $\mathcal{G}$ one can point a corresponding procedure representable as a term composed from the procedures corresponding to $\exists$–variables and $\vee$–branchings of the formula $\mathcal{F}$.*

Theorem 6 *The calculus J_c can be represented as a combination of calculus J' with some definite strategy ν restricting application of ω, i.e., any inference of $\mathcal{F} \Rightarrow \mathcal{G}$ in J_c corresponds to some inference of $\mathcal{F} \Rightarrow \mathcal{G}$ in $\langle J', \nu\rangle$ and vice–versa.*

Remark 2 The above mentioned justifies the soundness of the original calculus J usage (without transition to the calculus of J'–tasks) if the goal formula $\mathcal{G}$ is described in the class:

$$\exists \mathbb{T}\ \forall X : A\{\exists Y_1 : B_1, \ldots, \exists Y_n : B_n,\} \qquad (1)$$

and $\mathcal{F}$ is an arbitrary PCF, being even not a Horn one, i.e., by definition $\mathcal{F}$ has no branching in nodes with universal TQs. In this case reducing the task $\mathcal{F} \Rightarrow \mathcal{G}$ in J_c to the task $\exists \mathbb{T}\ \{\mathcal{F}, \bar{\mathcal{G}}\} \Rightarrow \exists \mathbb{F}$ in J' means, in turn, the possibility to replace the considered task by the proving the formula $\forall \mathbb{T}\ \exists \mathbb{T}\ \{\mathcal{F}, \bar{\mathcal{G}}\}$ in J (refutation of formula $(\mathcal{F})^* \& (\mathcal{G})^*$), i.e., next statement is hold.

Corollary 1 *Any task $\mathcal{F} \Rightarrow \mathcal{G}$ in J_c, where $\mathcal{G}$ is of the class (1), may be replaced by proving the formula $\forall \mathbb{T}\ \exists \mathbb{T}\ \{\mathcal{F}, \bar{\mathcal{G}}\}$ in the calculus J.*

On a problem of telescope guidance to the center of a planet in nonfull phase

In (Vassilyev & Cherkashin 1998) the problem of real-time intelligent guidance of telescope to the center of a planet in nonfull phase is considered as an application example. Usually it is realized by the consequence of measurements needed to locate telescope diaphragm concentrically to the bright limb of the planet (Estey 1968; Bilchenko, Matrosov, & Vassilyev 1973). The logical calculus allows us to extend the application area of control systems as compared with other approaches (Estey 1968; Bilchenko, Matrosov, & Vassilyev 1973).

In (Vassilyev & Cherkashin 1998) like in (Estey 1968; Bilchenko, Matrosov, & Vassilyev 1973) we use information on distances OH_i (Fig. 1) from the telescope center O to the bright limb of the planet in the focal plane of the telescope (the limb is represented by the arc AHB; the symmetry axis CH is perpendicular to the diameter AB). The aim of control: the center O has to track the center C of the planet (Fig. 1). To be more specific, the aim is to find values of the step engines signals (-1, 0, 1) coding the direction of error OC decrease. The current information for control synthesis consists of the distances $\rho_i = OH_i$ measured along the 8 scan directions a_i. So called ***informative bunch***, which consists of three neighbour scan directions intersecting the limb, is determined. The number of any informative bunch is equal to the number of its middle direction.

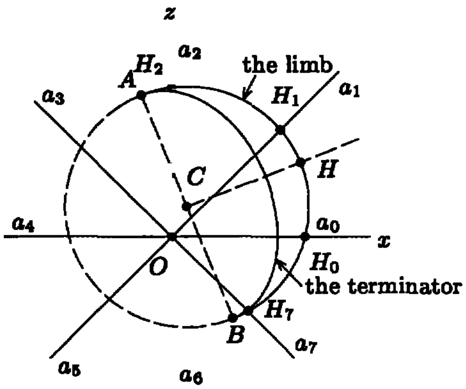

Fig. 1: *The planet image in focal plane xOz of telescope.*

The variables $r_1 = \mathrm{sgn}\,(\rho_{N\ominus 1} - \rho_N)$, $r_2 = \mathrm{sgn}\,(\rho_{N\oplus 1} - \rho_N)$ are formed, where $r_i \in \{-1, 0, 1\}$; operations $\oplus$, $\ominus$ are operations of addition and subtraction modulo 8. The planet disk is divided on areas in which the pair (r_1, r_2) has a corresponding fixed values. Each area has its own ***prescribed direction*** of error OC decrease. A recurrent identification of the area allows us to organize multistep process of guiding the planet center C by the center O. Thus, the control signals are completely computed by identified area and number i of the informative bunch for which the area is determined. But under the condition

$$\lambda < |\pi - \psi| \leq \frac{\pi}{2} \qquad (2)$$

there is a problem to find the informative bunch, i.e., it is necessary to distinct directions intersecting the limb and directions intersecting the terminator only. Here, ψ is a planet phase, $\lambda = \mathrm{const}$ is minimal sensible phase. The informative bunch always exists according to following condition:

$$\mathrm{arctg}\, q > 3\pi/8, \quad q = R/\max\sqrt{x_0^2 + z_0^2}, \quad q > 1, \qquad (3)$$

where $\max\sqrt{x_0^2 + y_0^2}$ is the maximal error of some rough guidance system used preliminary to locate the center C to satisfy (3). We suppose that the error of rough guidance is not more than $R/4$ (not more than one forth of the radius R).

In particular, it is clear that if along some direction a_i values ρ_i are less than $\frac{3}{4}R$ then a_i does not intersect the limb and intersects the terminator only. In this case, the informative bunch can be chosen as some "central" bunch among all $\rho_i \geq \frac{3}{4}R$ (actually, a bunch with the minimal number among them is chosen).

Thus, the set of logical rules of telescope control consist of four parts:

i) determination of informative bunch;
ii) determination of area where the center O is located;
iii) choosing direction to decrease the error OC;
iv) obtaining the control signals according to the direction chosen.

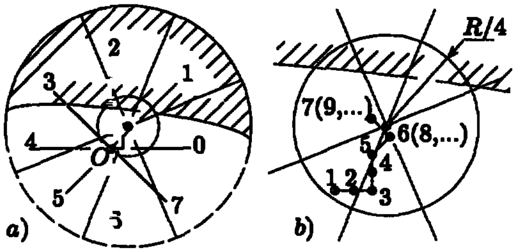

Fig. 2: *The example of guidance: starting position of the telescope center O in Fig. 2a corresponds to the point 1 in Fig. 2b.*

An example of telescope guidance process modeling is illustrated in Fig. 2. In the positions 1,2,3,4 of the point O the informative bunch 2 is used, and after that in the position 5 the scan direction a_0 begins to intersect the limb, and the bunch 1 is used. Starting from the point 6, the center O hits the ***insensibility area***, i.e., after this point the center O moves between the opposite areas where points 6,8,... are located and the area with points 7,9,... In this illustration we do not account for simplicity the dynamics of the planet with respect to the telescope base. Our intelligent control system possesses time–driven reasoning, since the knowledge is updated periodically and totally; the concrete values of data obtained out of measurements and calculations do not change the strategy of inference. That is why there is no necessity to use special temporal connectives or an extra variable for time as in (Gabbay & Reynolds 1995). If some improvements of the control is wanted the usage of the temporal reasoning will be beneficial.

Conclusion

In the progress of (Vassilyev 1990; Vassilyev & Zherlov 1995; Vassilyev 1997) the language of positively constructed formulas (PCFL) and its calculi are developed. The formula representation in the language has large structural elements with type quantifiers, but it is compact enough as compared with the clausal form used in resolution method, and, in addition, do not require scolemization and quantifiers elimination during conversion from the classical predicate calculus.

The logical calculi on the basis of PCFL is described in the paper, namely: descriptive calculus J, calculus of descriptive tasks J' and calculus of constructive tasks J_c. Some relations between them are shown. The only inference rule of J and its derivatives in J' and J_c deal with large sized items of the corresponding PCFL modification. This leads to essential reducing a search space during inference search. The PCFL and its calculi have good compatibility with heuristics.

The problem of intelligent guidance of telescope to the center of a planet in nonfull phase is considered briefly as an application example. As compared with (Bilchenko, Matrosov, & Vassilyev 1973) the developed method of control operates normally under extended set of admissible values of planet phase.

References

Åström, K. J., and McAvoy, T. J. 1992. *Handbook on Intelligent Control.* New York: Van Nostrand Reinhold. chapter "Intelligent Control: an Overview and Evaluation", 3–34.

Bilchenko, G.; Matrosov, V.; and Vassilyev, S. 1973. Guidance system of telescope to the planet center in nonfull phase. In *Proc. of IFAC Symp. on Automatic Control in Space.*

Butyrin, S., and et al. 1997. An expert system for design of spacecraft attitude control systems. *J. Artificial Intelligence in Engineering* (11):49–59.

Estey, R. 1968. *Automatic Control of Space Systems.* "Nauka" Publ. chapter "Device for Watching the Earth or the Moon in Nonfull Phase".

Gabbay, D. M., and Reynolds, M. 1995. *Handbook of Logic in Artificial Intelligence and Logic Programming,* volume 4, "Epistemic and Temporal Reasoning". Oxford: Clarendon Press. chapter "Towards a Computational Treatment of Time", 351–437.

Gabbay, D.; Hogger, C.; and Robinson, J., eds. 1994. *Handbook of Logic in Artificial Intelligence and Logic Programming,* volume 2, "Deduction Methodologies". Oxford: Clarendon Press.

Matrosov, V.; Sumenkov, S.; and Vassilyev, S. 1991. HEVROLOG — The intellectual integrating environment in system dynamics and control theory. In *Abst. of IMACS/IFAC Workshop. Methods and Software for Automatic Control Systems,* 106–108.

Patrushev, S., and Vassilyev, S. 1989. To automatization of VLSI structure synthesis. *Doklady Akad. Nauk* 306(2):301–304.

Sinha, N. K., and Gupta, M. M. 1996. *Intelligent Control Systems. Theory and Applications.* New York: IEEE Press. chapter "Toward intelligent machines", 804–807.

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

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

Vassilyev, S. N. 1990. Machine synthesis of mathematical theorems. *J. of Logic Programming* 9(2&3):235–266.

Vassilyev, S. N. 1996. Modeling logical derivation and hypothesis generation. In *Proc. of CESA'96 Conference,* volume 1, 148–153.

Vassilyev, S. 1997. A method for synthesis of the derivability conditions of the Horn and some other formulas. *Siberian Mathematical J.* 38(5):1034–1046.

White, D., and Sofge, D., eds. 1992. *Handbook on Intelligent Control.* New York: Van Nostrand Reinhold.