Annales Societatis Mathematicae Polonae Series IV:
Fundamenta Informaticae I - 1 (1977)

ALGORITHMIC LOGIC AND ITS APPLICATIONS IN THE THEORY OF PROGRAMS I

Grażyna Mirkowska

Abstract. The paper presents tools for formalizing and proving properties of programs. The language of algorithmic logic constitutes an extension of a programming language by formulas that describe algorithmic properties. The paper contains two axiomatizations of algorithmic logic, which are complete. It can be proved that every valid algorithmic property possesses a formal proof. An analogue of Herbrand theorem and a theorem on the normal form of a program are proved. Results of meta-mathematical character are applied to theory of programs, e.g. Paterson's theorem is an immediate corollary to Herbrand's theorem.

EFFECTIVITY PROBLEMS OF ALGORITHMIC LOGIC

Antoni Kreczmar

Abstract. In the paper we solve some effectivity problems of program schemas. Such properties of programs as, for example, the strong and the weak equivalence, the correctness and the partial correctness of a program, the halting problem are classified in Kleene-Mostowski hierarchy. A basic tool used in the paper is algorithmic logic.

A NATURAL METHOD OF PROVING PROPERTIES OF PROGRAMS

Józef Winkowski

Abstract. In this paper an approach to proving properties of programs is presented, which slightly differs from the methods developed up to now. In contrast. to other approaches it needs no formal program notion but only a formalization of some basic program properties.
The key idea in this approach is to associate with any program under consideration a formal theory in such a way that all the processes, which may be thougth as possible executions of the program considered, become models of the associated. theory. Such a theory can be constructed by formulating axioms about the program executions, and about quantities which explicitly or implicitly occur in the program and change their values while the program is being executed. Appropriate axioms and the properties which are to be proved can easily be described by formulas of the predicate logic. This enables one to consider various properties of the program as theorems of the resulting theory and to prove them by usual means of the predicate logic.

THREE-QUERIES FILE ORGANIZATIONS

Witold Lipski, Jr.

Abstract. The following combinatorial problem arising in file organization is considered: Given three sets $M_1, M_2, M_3 \subseteq X$, find an arrangement of X such that each M_i is a segment of consecutive elements of X. Also some theorems extending the results of Ghosh concerning the two-dimensional consecutive retrieval property are proved. In particular, necessary and sufficient condition for a family of three sets to admit a two-dimensional organization is given.

ON MAXIMAL PARALLEL REALIZATION OF PROCESSES

Piotr Dembiński *61-69*

Abstract. A simple (finite automata) model for parallel computation is considered. The parallelism is replaced by nondeterminism. The class of all parallel realizations of given processes is algebraically characterized, and a natural "less parallel" gelation between realizations is introduced. A sufficient and necessary condition is riven for the existence of the maximally parallel realization in the set of correct realizations (distinguished by means of computational invariants).

M-GROUPOID AS A TOOL FOR INVESTIGATING MATHEMATICAL MODELS OF COMPUTERS

Jerzy Tiuryn *71-91*

Abstract. An M-groupoid is a simplified model of computer. The classes of M-groupoids, address machines, stored program computers and iterative systems are presented as categories - by a suitable choice of homomorphisms. It is shown that the first three categories are equivalent, whereas the fourth is weaker (it is not equivalent to the previous ones and it can easily be embedded in the category of M-groupoids). This fact proves that M-groupoids form an essentially better and reasonably simple approximation of more complicated models of computers than iterative systems.

INVESTIGATIONS OF PROPERTIES OF PROGRAMS BY MEANS OF THE EXTENDED ALGORITHMIC LOGIC I

Lech Banachowski *93-119*

Abstract. The present paper contains investigations concerning the semantic correctness of programs. Presented methods of analysis of programs are appropriate for every domain of computation. Algorithmic logic extended by classical quantifiers is a fundamental mathematical tool used in the paper. Interrelations between properties of programs and properties of descriptions of programs are studied (a description of a program is a mathematical model of the notion of a documentation of a program).

STOCHASTIC INFORMATIONAL SYSTEMS I

Wiktor Marek, Tadeusz Traczyk *121-130*

Abstract. In the paper the notion of a stochastic informational retrieval sys. tens is introduced and discussed. This is a generalization of the notion of an i.s.r. system in the case of data which are measured with some precision. The method of numerical Boolean algebras is applied. Theorems connected with the decomposition of such systems are proved.

AN ALGEBRAIC APPROACH TO THE THEORY OF RECURSIVE COROUTINES

Ryszard Janicki *131-145*

Abstract. A mathematical model for recursive coroutines in introduced. Relationships between Mazurkiewicz algorithms and this model are considered. Some linguistic and computational properties of coroutines are proved. The set of all functions computed by programs with recursive coroutines is proved to contain the set of all functions computed by commonly used recursive programs.

Annales Societatis Mathematicae Polonae Series IV:
Fundamenta Informaticae I (1977)

Abstract. In the present paper we investigate algorithmic properties of fields. We prove that axioms of formally real fields for the field ***R*** of reals and axioms of fields of characteristic zero for the field ***C*** of complex numbers, give the complete characterization of algorithmic properties. By Kfoury's theorem programs which define total functions over ***R*** or ***C*** are effectively equivalent to loop-free programs. Examples of programmable and nonprogrammable functions and relations over ***R*** and ***C*** are given. In the case of ordered reals the axioms of Archimedean ordered fields completely characterize algorithmic properties. We show how to use the equivalent version of Archimed's axiom (the exhaustion rule) in order to prove formally the correctness of some iterative numerical algorithms.

Keywords: Programs and programmability, algorithmic properties, programmability in fields, axioms for algorithmic properties of reals, ordered reals and complex numbers.

Abstract. There are a great many research works concerning the well-known stochastic automata of Moore, Mealy, Rabin, Turing and others. Recently an automaton of Markov's chain type has been introduced by Bartoszyński. This automaton is obtained by a generalization of Pawlak's deterministic machine.

The aim of this note is to give a concept of a stochastic automaton of Markov's generalized chain type. The introduced automaton called a stochastic, k-automaton (s.k-a.) is a common generalization of Bartoszyński's automaton and Grodzki's deterministic k-machine. By a stochastic k-automaton we mean an ordered triple $M_k =< U, a, \pi >, k \geq 1$, where U denotes a finite nonempty set, a is a function from U^k to $[0,1]$ with $\sum_{v \in U^k} a(v) = 1$, and π is a function from U^{k+1} to $[0,1]$ with $\sum_{u \in U} \pi(v,u) = 1$ for every $v \in U^k$.

For all $N \geq k$ we can define a probability measure P_N on $U^N = U \times U \times \ldots \times U$ as follows: $P_N(u_1, u_2, \ldots, u_N) = a(u_1, u_2, \ldots, u_k)\pi(u_1, u_2, \ldots, u_{k+i})\pi(u_2, u_3, \ldots, u_{k+2}) \ldots \pi(u_{N-k}, u_{N-k+1}, \ldots, u_N)$.

We deal with the problems of the shrinkage and the extension of a system of s.k-a.' s $M_k^{(i)} =< U, a^{(i)}, \pi^{(i)} >$, $i = 1, 2, \ldots, m, m \geq 2$. In this note there are given conditions under which an s.k-a. $M_k =< U, a, \pi >$ exists and the language of this automaton defined as $L_M = \{(u_1, u_2, u_3, \ldots) : \bigwedge_{N \geq 1} P_N(u_1, u_2, \ldots, u_N) > 0\}$ either contains the languages of all the automata $M_k^{(i)}$, $i = 1, 2, \ldots, m$, or this language equals the intersection of all those languages.

Keywords: Stochastic k-automaton, extension, shrinkage, N-word, set of N-words, words, language, probability measure, carrier, concordance, truly concordance, pairwise concordance.

SETS OF SEQUENCES CONSTRUCTED BY NONDETERMINISTIC k-MACHINES AND DETERMINISTIC k-MACHINES

Jiří Karasek *243-250*

Abstract. The paper deals with nondeterministic and deterministic k-machines which represent, in a certain sense, generalizations of some well-known notions. Connections between the sets of sequences constructed by nondeterministic and deterministic k-machines, and some other related problems are investigated.

Keywords: Deterministic machines, nondeterministic machines, finite automata.

PPC (PISA PROOF CHECKER): A TOOL FOR EXPERIMENTS IN THEORY OF PROVING AND MATHEMATICAL THEORY OF COMPUTATION

Luigia Aiello, Mario Aiello, Giuseppe Attardi, Gianfranco Prini *251-275*

Abstract. An interactive proof checker is a system which is able of building a formal proof (in some deductive calculus) by executing commands provided by the user. Proof checkers arc useful both for making experiments in proof construction within various formal systems and for proving theorems in those fields of mathematics (such as mathematical theory of computation) where proofs are necessarily very large and unfeasible by hand. Two levels may be distinguished in a proof checker. The lower one implements the proof management routines, and is independent of any particular logic. The higher one implements the inference rules of a particular logical calculus. Powerful higher level rules are also needed to make the use of the checker practical. Almost all routine steps may be then generated automatically, and the user has just to give some "hints" to the checker, which transforms an "informal argument" into a formal proof.

Keywords: Automatic theorem proving, denotational semantics of programming languages, mathematical theory of computation, proof of formal properties of programs.

ERRATA TO VOLUME I NUMBER 1

page, line:	instead of:	should be:
5_{15}	$\hat{v}(a_i) = \tau_{iR}(v)$	$\hat{v}(a_i) = a_{iR}(v)$
9^{4}	$\tau_R(v) = \chi(\tau)$	$\tau_R(v) = \chi(\tau)_R(v)$
12^{10}	$\underline{\vee}[aK[]]_R^{i_0}(v)$	$\underline{\vee}[aK[]]_R^{j_0}(v)$
13^{10}	$a_R(v) = \mathbf{0}$	$a_R(v) = 1$
13_{9}	$x))\}_{i\in\mathcal{N}}$	$x))\}_{i\in I}$
15_{1}	$\underline{\vee}[\delta K[]] \neg (\delta \cap a)$	$\underline{\vee}[\delta K[]](\neg\delta \cap a)$
61^{8}	rizations	lizations
61^{9}	gelation	relation
61^{10}	riven	given
61^{11}	lealizations	realizations
cover 3^{9}	pronters	printers
cover 3_{2}	Concurrens	Concurrent

Annales Societatis Mathematicae Polonae Series IV:
Fundamenta Informaticae I (1978)

Abstract. The aim of this paper is to explain what it means to simulate a process in a computer. The main observation is that any computer simulation reduces to modelling a theory of the process which is to be simulated in a theory of a computation process. This makes it possible to formulate and solve various problems concerning the correctness of simulation programs.
An example is given to illustrate certain details of the presented point of view.

Keywords: Process, theory, interpretation, modelling.

Abstract. We give here a general method of searching out the proofs in, any formalized. theory based on an enumerable language. We discuss for which hind. of theories this method is effective.

Keywords: Invertible, completely invertible machine, proof, theorem proving system.

Abstract. The purpose of this paper is to give a full description of the semantics of the language ALGOL. In this way we would like to illustrate a new idea of the semantics of a program with block structures and procedures.
The definition of the syntax of language ALGOL 1 (which is a restriction of the language ALGOL 60) is given. A full description of the semantics of the language ALGOL 1 is given and the notion of a computation of a program is introduced and illustrated by two examples. The following theorem is proved: for some expressions ω and for every substitutions $s : z := \omega$ there exists a finite sequence of substitutions $s_i : z_i := \omega_i$ which is equivalent to s and such that in every expression ω_i at most one functor occurs.

Keywords: Programming language syntax, programming language semantics, program with block structures, realization of a programming language, valuation, computation of a program.

Abstract. The relationship between. ETOL systems and context-free programmed grammars is investigated. The characterization of the class of ETOL languages, and its main subclasses (EOL, EDTOL, EDOL) in terms of various subclasses of the class of context-free programmed grammars is provided. Then an extension of the notion of an ETOL system is investigated. In this way a characterization of the class of context-free programmed languages is provided.

Keywords: Formal languages, L-systems, context-free programmed grammars.

AN ELIMINATION OF ITERATION QUANTIFIERS IN A CERTAIN CLASS OF ALGORITHMIC FORMULAS

Grażyna Mirkowska, Ewa Orłowska *347-355*

Abstract. The paper presents a method for elimination of iteration quantifiers from some type of formulas of algorithmic language. This is a first step to construct in algorithmic logic a method of automatic proving theorems. The presented approach enables us to investigate open algorithmic theories.

Keywords: Algorithmic logic, program, procedure.

ON SIMPLE CONTROLLED PROGRAMS

Wojciech Rytter *357-363*

Abstract. The paper presents augmented one-variable programs. Four problems concerning such programs are introduced and examined by using concepts of the theory of automata. Some applications of the automata theory to the theory of programming are given.

Keywords: Programs, simple controlled programs, nondeterministic automaton, equivalence, synchronizability, halting.

RECURSIVELY ENUMERABLE DEGREES OF PROGRAMMING PROBLEMS

Andrzej Włodzimierz Mostowski *365-377*

Abstract. The paper deals with r.e. degrees of unsolvability of the halt problem and the equivalence problem. Programs with very few programming aids are investigated.
The connections between degrees of unsolvability of the halt problem for a family of programs as a whole and its members are investigated. The operations nonincreasing the degree of the halt problem are investigated. Some connections between the equivalence and the halt problems for a family are given. For precise formulation of main results cf. Theorems 1-5.

Keywords: Theory of programs, halt problem, degrees of unsolvability, equivalence of programs, structured programming.

ALGORITHMIC PROPERTIES OF PROGRAMS WITH TABLES

Wiktor Dańko *379-398*

Abstract. In this paper we consider a formal language of programs with tables containing also formulas describing properties of programs. Our main aim is to characterize the set of axioms and the set of rules of inference such that a formula a is a tautology if and only if a can be obtained by means of rules of inference from axioms.

Keywords: Algorithmic logic, programs with tables, algorithmic logic with tables, algorithmic properties, tautology, completeness property.