13 (1) 1990

Editor's Foreword

Krzysztof R. Apt *v-ix*

ARITHMETIC CLASSIFICATION OF PERFECT MODELS OF STRATIFIED PROGRAMS

Krzysztof R. Apt, Howard A. Blair *1-17*

Abstract. We study here the recursion theoretic complexity of the perfect (Herbrand) models of stratified logic programs. We show that these models lie arbitrarily high in the arithmetic hierarchy. As a byproduct we obtain a similar characterization of the recursion theoretic complexity of the set of consequences in a number of formalisms for non-monotonic reasoning. We show that under some circumstances this complexity can be brought down to recursive enumerability.

STRATIFIED, WEAK STRATIFIED, AND THREE-VALUED SEMANTICS

Melvin Fitting, Marion Ben-Jacob *19-33*

Abstract. We investigate the relationship between three-valued Kripke/Kleene semantics and stratified semantics for stratifiable logic programs. We first show these are compatible, in the sense that if the three-valued semantics assigns a classical truth value, the stratified approach will assign the same value. Next, the familiar fixed point semantics for pure Horn clause programs gives both smallest and biggest fixed points fundamental roles. We show how to extend this idea to the family of stratifiable logic programs, producing a semantics we call *weak stratified*. Finally, we show weak stratified semantics coincides exactly with the three-valued approach on stratifiable programs, though the three-valued version is generally applicable, and does not require stratification assumptions.

SOME REMARKS ON THE COMPLETED DATABASE

Kenneth Kunen *35-49*

Abstract. We show how Clark's Completed Database (CDB) can be extended to include a number of Prolog constructs usually considered to be outside of pure logic, while still maintaining the known partial completeness results for SLDNF derivations. We also prove that in the semantics provided by the CDB and SLDNF, there is no definition of transitive closure which is strict and does not use function symbols.

WEAKLY STRATIFIED LOGIC PROGRAMS

Halina Przymusinska, Teodor C. Przymusinski *51-65*

ON DOWNWARD CLOSURE ORDINALS OF LOGIC PROGRAMS

Rajiv Bagai, Marc Bezem, and M.H. Van Emden *67-83*

Abstract. Blair has shown that for every ordinal up to and including the least non-recursive ordinal there exists a logic program having that ordinal as downward closure ordinal. However, given such an ordinal and Blair's proof, it is not straightforward to find a corresponding logic program. In fact, in the literature only a few isolated, *ad hoc*, examples of logic programs with downward closure ordinal greater than ω can be found. We contribute to bridging the gap between what is known

abstractly and what is known concretely by showing the connection between some of the existing examples and the well-known concept of the order of a vertex in a graph. Using this connection as a basis, we construct a family $\{P_\alpha\}_{\alpha<\varepsilon_0}$ of logic programs where any member P_α, has downward closure ordinal $\omega + \alpha$.

13 (2) 1990

A PROCESS SPECIFICATION FORMALISM

S. Mauw, and G.J. Veltink *85-139*

Abstract. Traditional methods for programming sequential machines are inadequate for specifying parallel systems. Because debugging of parallel programs is hard, due to e.g. non-deterministic execution, verification of program correctness becomes an even more important issue. The Algebra of Communicating Processes (ACP) is a formal theory which emphasizes verification and can be applied to a large domain of problems ranging from electronic circuits to CAM architectures. The manual verification of specifications of small size has already been achieved, but this cannot easily be extended to the verification of larger industrially relevant systems. To deal with this problem we need computer tools to help with the specification, simulation, verification and implementation. The first requirement for building such a set of tools is a specification language. In this paper we introduce PSF_d (Process Specification Formalism - draft) which can be used to formally express processes in ACP. In order to meet the modern requirements of software engineering, like reusability of software, PSF_d supports the modular construction of specifications and parameterization of modules. To be able to deal with the notion of data, ASF (Algebraic Specification Formalism) is embedded in our formalism. As semantics for PSF_d a combination of initial algebra semantics and operational semantics for concurrent processes is used. A comparison with programming languages and other formal description techniques for the specification of concurrent systems is included.

A COMPOSITIONAL SEMANTICS FOR TIMED PETRI NETS

Andrea Maggiolo-Schettini, and Jozef Winkowski *141-170*

Abstract. Timed Petri nets and their behaviours are considered. A concept of a seminet is introduced, which generalizes the concept of a net, and suitable operations on seminets are defined, which allow constructing seminets from atoms corresponding to places and transitions. The behaviours of seminets are given in the form of so called configuration systems, a notion close to labelled event structures. Such behaviours can be combined with the aid of operations corresponding to those on seminets. In particular, the behaviour of a compound seminet can be obtained by combining the behaviours of components.

Keywords: timed Petri net, seminet, composition, hiding, synchronization structure, configuration, timed configuration system, delayed configuration, eager configuration system, eager kernel.

TOWARDS A NEW ALGEBRAIC FOUNDATION OF FLOWCHART SCHEME THEORY

Virgil Emil Căzănescu, and Gheorghe Ştefănescu *171-210*

Abstract. We develope a formalism for the algebraic study of flowchart schemes and their behaviours, based on a new axiomatic looping operation, called feedback.

This formalism is based on certain flownomial expressions. Such an expression is built up from two types of atomic schemes (i.e., elements in a double-ranked set X considered as unknown computation processes, and elements in a "theory" T considered as known computation processes) by using three operations: sum, composition, and feedback. Flownomial expressions are subject to certain rules of identification. The axiomatization of flowchart schemes is based on the fact that a flowchart scheme may be identified with a class of isomorphic flownomial expressions in normal form. The corresponding algebra for flowchart schemes is called biflow.

This axiomatization is extended to certain types of behaviour. We present axiomatizations for accessible flowchart schemes, reduced flowchart schemes, minimal flowchart schemes with respect to the input behaviour, minimal flowchart schemes with respect to the input-output behaviour etc. Some results are new, others are simple translations in terms of feedback of previous results obtained by using Elgot's iteration or Kleene's repetition.

The paper also contains some historical comments.

A CHARACTERIZATION OF IRREDUCIBLE SETS MODULO LEFT-L1NEAR TERM REWRITING SYSTEMS BY TREE AUTOMATA

Z. Fülop, and S. Vágvölgyi *211-226*

Abstract. The concept of top-down tree automata with prefix look-ahead is introduced. It is shown that a tree language is the set of irreducible trees of a left-linear term rewriting system if and only if it can be recognized by a one-state deterministic top-down tree automaton with prefix look-ahead.

A NOTE ON ROUGH CONCEPTS LOGIC

Michal Krynicki *227-235*

Abstract. In papers [4,5] Pawlak introduced the notion of a rough set and approximation space. In [6] Pawlak formulated some concept of rough logic. The notion of the approximate truth was considered by many philosophers and logicians and in the last time by computer scientists. This was motivated by some research in artificial intelligence as for example expert systems, approximate reasoning methods and information system with imprecise information.

The concept of rough logic introduced in [6] based on the notion of approximate truth determined by rough sets. Following these ideas Rasiowa and Skowron in [7] proposed the a.propriate first order logic for concepts of rough definability. We denote this logic by L_R. In [9] Szczerba proposed some logic with additional quantifier as rough concepts logic. We denote this logic by $L(Q_R)$. The aim of this paper is a comparizing of these two logics with respect to their expressive power and giving some propositions of some modificated versions of rough concepts logics.

We use more or less standard notation. By $[a]_R$ we denote the equivalence class of the element a with respect to the equivalence relation R. We write $L \leq L'$ if expressive power of the logic L is weaker then the expressive power of the logic 1,1 (i.e. each class of models definable by a sentence from L is also definable by a sentence from L'). If $L \leq L'$ and $L' \leq L$ then we say that L and L' are equivalent and denote this by $L \equiv L'$. If $L \leq$ but $L \not\equiv L'$ then we write $L < L'$.

13 (3) 1990

REDUCING OPERATORS FOR GENERALIZED GRAMMARS???

Miroslav Novotný *237-244*

Abstract. Theory of reducing operators elaborated for pure generalized grammars in [5] is transferred to normed generalized grammars. There exists a reducing operator ϱ such that for any normed generalized grammar G the norrned generalized grammars $G, \varrho(G)$ generate the same languages; furthermore, $\varrho(G)$ is a normed grammar if and only if there exists a normed grammar G' such that G and G' generate the same language and G' is in a certain sense smaller than or equal to G. The reducing operator ϱ is used to characterize the so called harmonic languages.

Keywords: normed generalized grammar, permitting triple defined by means of derivatives, language grammatizable by means of derivatives, language weakly grammatizable by means of derivatives, harmonic grammar, harmonic language, reducing operator for generalized grammars.

THE ROUGH SETS THEORY AND EVIDENCE THEORY

Andrzej Skowron *245-262*

Abstract. The aim of the paper is to show some connections between the rough sets theory and the Dempster-Shafer approach. We prove that for every Pawlak's approximation space there exists a Dempster-Shafer space with the qualities of the lower and upper approximations of sets in the approximation space equal to the credibility and plausibility of sets in the Dempster-Shafer space, respectively.

Analogous connections hold between approximation spaces generated by the decision tables and Dempster-Shafer spaces, namely for every decision table space there exists a Dempster-Shafer space such that the qualities of the lower and upper approximations C with respect to the condition attributes) of sets definable in the decision table by condition and decision attributes coincide with the credibility and plausibility of sets in the Dempster-Shafer space, respectively.

A combination rule in approximation spaces analogous to the combination rule used in the Dempster approach is derived.

COMMUNICATING PROCESSES AND ENTROPIC ALGEBRAS

Anna B. Romanowska, and Jonathan D.H. Smith *263-274*

Abstract. A significant contribution to the analysis of certain aspects of the communicating processes model was made by D. Benson's proposal to view incompletely specified nondeterministic processes as modules over certain semirings, and dually as comodules over corresponding coalgebras. The effectiveness of the proposal in treating the synthesis of such processes under mutual communication depended on the good behaviour of these algebraic systems with respect to tensor products. The aim of the paper is to draw attension to the algebraic theory underlying Benson's proposal, the theory of entropic algebras. Working with entropic algebras guarantees that tensor products are sufficiently well-behaved to make Benson's theory work.

VECTOR CONTROLLED CONCURRENT SYSTEMS PART I: BASIC CLASSES

N.W. Keesmaat, H.C.M. Kleijn, and G. Rozenberg *275-316*

Abstract. A model of concurrent systems, called Vector Controlled Concurrent Systems, is introduced. It generalizes the vector synchronization mechanism used in path expressions (COSY). In

this first part of the paper, which consists of two parts, the basic model is introduced and motivated, and then a distinction is made between static and dynamic submodels.

ALGEBRAIC FOUNDATIONS OF LOGIC PROGRAMMING, I: THE DISTRIBUTIVE LATTICE OF LOGIC PROGRAMS

Anil Hirani, and V.S. Subrahmanian *317-332*

Abstract. Given a logic program P, the operator T_p associated with P is closely related to the intended meaning of P. Given a first order language L that is generated by finitely many non-logical symbols, our aim is to study the algebraic properties of the set $\{T_p | P$ is a general logic program in language $L\}$ with certain operators on it. For the operators defined in this paper the resulting algebraic structure is a bounded distributive lattice. Our study extends (to the case of general logic programs), the work of Mancarella and Pedreschi who initiated a study of the algebraic properties of the space of pure logic programs. We study the algebraic properties of this set and identify the ideals and zero divisors. In addition, we prove that our algebra satisfies various *non-extensibility* conditions.

RESOLUTION IN THE DOMAIN OF STRONGLY FINITE LOGICS

Zbigniew Stachniak, and Peter O'Hearn *333-351*

Abstract. In this paper the notion of a resolution counterpart of a propositional logic is introduced and studied. This notion is based on a generalization of the resolution rule of J.A. Robinson. It is shown that for every strongly finite logic a refutationally complete nonclausal resolution proof system can be constructed and that the completeness of such systems is preserved with respect to the polarity and set of support strategies.

INFINITARY BEHAVIOURS AND INFINITARY OBSERVATIONS

Philippe Darondeau, and Boubakar Gamatie Irisa *353-386*

Abstract. We present a denotational model for CCS, accounting for finite as well as infinite behaviours. The meanings of programs are combinations of ready sets and infinitary languages. The model is shown fully abstract w.r.t. the equivalence $u \sim v$ if $L(u,t) = L(v,t)$ for any testing program t, where $L(p,q)$ is the set of totally ordered traces of communications between programs p and q set in parallel.

TOWARDS AVERAGE COMPLEXITY OF PROPOSITIONAL BINARY PROLOG PROGRAMS

Hans Kleine Böning, and Ulrich Löwen *387-399*

Abstract. We investigate the average complexity of various classes of binary propositional Prolog programs. We consider classes of tree-like programs introducing a concept of unary and binary edges in a tree to manage situations where a consequence $A \leftarrow B$ of a Prolog program can be derived in several ways. We establish exponential lower bounds for the average complexity of tree-like Prolog programs having no facts and we have polynomial upper bounds for various classes of tree-like Prolog programs with facts. These results should be considered as a first step towards the average complexity of propositional Prolog programs.

13 (4) 1990

GUEST EDITOR'S NOTE.

Wiktor Marek *401-401*

FORMALIZATION OF INHERITANCE REASONING IN AUTOEPISTEMIC LOGIC

Michael Gelfond, and Halina Przymusinska *403-443*

Abstract. Current research in the area of nonmonotonic reasoning suggests that autoepistemic logic provides a general framework for formalizing commonsense reasoning in various domains of discourse. The goal of this paper is to investigate the suitability of autoepistemic logic for formalization of some forms of inheritance reasoning. To this end we propose a new semantics for inheritance networks with exceptions based on autoepistemic logic.

WELL-FOUNDED SEMANTICS COINCIDES WITH THREE VALUED STABLE SEMANTICS

Teodor Przymusinski *445-463*

Abstract. We introduce *3-valued stable models* which are a natural generalization of standard (2-valued) *stable models*. We show that every logic program P has at least one 3-valued stable model and that the *well-founded model* of any program P [Van Gelder et al., 1990] coincides with the *smallest 3-valued stable model* of P. We conclude that the well-founded semantics of an arbitrary logic program *coincides* with the 3-valued stable model semantics.

The 3-valued stable semantics is closely related to non-monotonic formalisms in AI. Namely, every program P can be translated into a suitable autoepistemic (resp. default) theory $\hat{P}$ so that the 3-valued stable semantics of P coincides with the (3-valued) autoepistemic (resp. default) semantics of $\hat{P}$. Similar results hold for circumscription and CWA. Moreover, it can be shown that the 3-valued stable semantics has a natural extension to the class of all disjunctive logic programs and deductive databases.

Finally, following upon the recent approach developed by Gelfond and Lifschitz, we extend all of our results to more general logic programs which, in addition to the use of *negation as failure*, permit the use of *classical negation*.

YLOGIC: A FRAMEWORK FOR REASONING ABOUT CHAMELEONIC PROGRAMS WITH INCONSISTENT COMPLETIONS

V.S. Subrahmanian *465-483*

Abstract. Large logic programs are normally designed by teams of individuals, each of whom designs a subprogram. While each of these subprograms may have consistent completions, the logic program obtained by taking the union of these subprograms may not. However, the resulting program still serves a useful purpose, for a (possibly) very large subset of it still has a consistent completion. We argue that "small" inconsistencies may cause a logic program to have no models (in the traditional sense), even though it still serves some useful purpose. A semantics is developed in this paper for general logic programs which ascribes a very reasonable meaning to general logic programs irrespective of whether they have consistent (in the classical logic sense) completions.

ON THE CLASSIFICATION AND EXISTENCE OF STRUCTURES IN DEFAULT LOGIC

Aidong Zhang, and Wiktor Marek *485-499*

Abstract. We investigate possible belief sets of an agent reasoning with default rules. Besides of Reiter's extensions which are based on a proof-theoretic paradigm (similar to Logic Programming), other structures for default theories, based on weaker or different methods of constructing belief sets are considered, in particular, weak extensions and minimal sets. The first of these concepts is known to be closely connected to autoepistemic expansions of Moore, the other to minimal stable autoepistemic theories containing the initial assumptions. We introduce the concept of stratifed collection of default rules and investigate the properties of the largest stratified subset of the family D, determined by W. We find a necessary and sufficient condition for a weak extension to be an extension in terms of stratification. We prove that for theories (D, W) without extension, the least fixed point of the associated operator (with weak extension or minimal set as a context) is an extension of suitably chosen (D', W) with $D' \subseteq D$. We investigate conditions for existence of extensions and introduce the notion of perfectly-stratified set of default rules and its variant of maximally perfectly-stratified set. Existence of such set of default rules turns out to be equivalent to the existence of extension. Finally, we investigate convergence of algorithm for computing extensions.