Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

452

B. Rovan (Ed.)

MFCS '90

Mathematical Foundations of Computer Science 1990

Banská Bystrica, Czechoslovakia
August 27–31, 1990
Proceedings

Springer-Verlag
Berlin Heidelberg New York London
Paris Tokyo Hong Kong Barcelona

Editor

Branislav Rovan
Department of Computer Science
Comenius University
842 15 Bratislava, Czechoslovakia

CR Subject Classification (1987): F.1–4, G.2, D.3.1, D.4.1, E.5, I.1.2

ISBN 3-540-52953-5 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-52953-5 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

FOREWORD

This volume contains papers selected for presentation at the 15th Symposium on Mathematical Foundations of Computer Science - MFCS'90, held in Banská Bystrica, Czechoslovakia, August 27-31, 1990.

It is the fifteenth symposium in the series of international meetings which took place in Czechoslovakia and Poland. The aim of these symposia is to bring together specialists in theoretical fields of computer science from various countries and to stimulate mathematical research in theoretical computer science. The previous meetings took place in Jablona, 1972; Štrbské pleso, 1973; Jadwisin, 1974; Mariánské Lázně, 1975; Gdańsk, 1976; Tatranská Lomnica, 1977; Zakopane, 1978; Olomouc, 1979; Rydzina, 1980; Štrbské Pleso, 1981; Prague, 1984; Bratislava, 1986; Carlsbad, 1988; and Porąbka-Kozubnik, 1989.

The Proceedings consist of invited papers and communications. The latter have been selected by the international Program Committee from a total of 128 submitted papers. The following program committee members took part in the evaluation and the selection of submitted papers: S.Abiteboul (Paris), D.Bjørner (Lyngby), M.Broy (München), M.Chytil (Prague), P. van Emde Boas (Amsterdam), M.Fischer (New Haven), I.Guessarian (Paris), O.Ibarra (Santa Barbara), P.Kanellakis (Providence), I.Korec (Bratislava), V.E.Kotov (Novosibirsk), K.Mehlhorn (Saarbrücken), G.Mirkowska (Warsaw), D.Mundici (Milan), I.Németi (Budapest), B.Rovan - chairman (Bratislava), Ch.Rackoff (Toronto), A.Salomaa (Turku), G.Wechsung (Jena).

I would like to thank all the Program Committee members for their meritorious work in evaluating the submitted papers as well as to the following referees, who assisted the Program Committee members: S. M. Achasova, H. Andréka, J. M. Autebert, E. Badouel, L. Banachowski, O. L. Bandman, W. Bartol, D. Beauquier - Girault, J. Beauquier, R. Beigel, U. Berger, R. Berghammer, G. Bernot, J. Berstel, A. Bertoni, M. Bidoit, L. Boasson, H. L. Bodlaender, J. D. Boissonnat, A. Bossi, Bouge, M. Breu, L. Brim, L. Bukovský, M. A. Bulyonkov, A. V. Bystrov, K. Chen, L. A. Cherkasova, C. Choffrut, C. Choppy, J.-L. Coquidé, R. Cori, K. Culik II., P. Darondeau, J. Dassow, F. Dedericks, J. P. Delahaye, P. Dembiński, G. Demichelis, J. Deminet, C. Dendorfer, J. Desel, A. Deutsch, M. Dezani, C. Dony, V. Donzeau - Gouge, A. G. Dragalin, P. Ďuriš, P. Enjalbert, L. Fariñas del Cerro, P. Feautrier, M. Ferbus, M. Ferenczi, A. S. Filurin, Ch. Frougny, U. Furbach, J. Gallier, D. Gardy, P. Gastin, V. Geffert, A. Geser, F. Gire, O. Goldreich, M. Goldwurm, S. Grigorieff, J. Gruska, M. R. Hansen, A. E. Haxthausen, T. Harju, R. Hennicker, A. Hemmerling, D. Hillen, P. Jacquet, P. Jančar, K. P. Jantke, C. Jard, R. Jeansoulin, K. M. Jensen, T. Jiang, J.-P. Jouannaud, H. Jung, J. Karhumäki, J. Kari, T. Katriňák, A. Kelemenová, H. C. Kirchner, J. Kivinen, J. Komara, V. Koubek, M. Krause, F. Kröger, M. Křivánek, G. Kucherov, G. A. Lanzarone, M. J. Lao, M. Latteux, J.J. Levy, G. Linde, M. Linna, G. Lischke, I. Litovsky, A. I. Litwiniuk, H. H. Lovengreen, W. Lukaszewicz, A. Martelli, G. Mauri, Ch. Meinel, D. Meinhardt, D. Mery, T. Mihálydeák, M. Milkowska, M. Mňuk, B. Möller, H. Müller, F. Nickl, M. Nielsen, V. Niemi, D. Niwinski, E. Ochmanski, R. Ochranová, E. Orlowska, P. Padawitz, M.A. Palis, A. Peht, M. Pellentier, D. Perrin, A. Petit, J. Plesník, L. Pomello, L. Priese, I. Prívara,

P. Pudlák, T. Puls, C. Queinnec, J. C. Raoult, B. Ravikumar, H. Rischel, Ph. Robert, L. Rosier, G. Rozenberg, P. Ružička, W. Rytter, J. Sakarovitch, A. Salwicki, A. Saoudi, P. Savický, B. A. Schieder, H. Schlingloff, G. Schmidt, M. Scholl, H. Schröder, D. Schupp, G. Serény, V. N. Shilov, I. Sifakis, C. Simone, A. Slobodová, J. Stavnstrup, J. Steensgaard - Madsen, M. Steinby, J. Steiner, O. Sýkora, A. Szalas, Szécsényi, Gy. Szérény, M. Szöts, J. Šefránek, M. Škoviera, J. Šturc, V. Tannen, A. Tarlecki, J. Tiuryn, D. Uhlig, E. Ukkonen, S. Valentini, D. Vidal, J. Villadsen, L. Voelkel, J. Vogel, I. Vrťo, S. Waack, K. W. Wagner, S. Waligórski, J. Warpechowska, R. Weber, J. Westbrook, J. Wiedermann, M. Wieth, M. Wirsing, A. Zanardo, T. Zeugmann, W. Zielonka, J. Zlatuška.

MFCS'90 was organized by the Association of Slovak Mathematicians and Physicists of the Slovak Academy of Sciences, The House of Technology in Banská Bystrica, and the Comenius University in Bratislava in cooperation with the following institutions: VUSEI-AR, Bratislava; Computing Center of the Slovak Academy of Sciences, Bratislava; Masaryk University, Brno; Šafárik University, Košice; Charles University, Prague; Slovak Technical University, Bratislava.

The Organizing Committee consisted of G. Andrejková, E. Bitnerová, Z. Bouša, I. Černá, Z. Felixová, R. Galbavý, J. Hromkovič, J. Hvorecký, V. Jankovič, B. Rovan, P. Ružička, D. Senčeková, A. Slobodová, H. Štefanová, I. Vrťo.

Being the editor of these Proceedings I am much indebted to all contributors to the scientific program of the symposium, especially to the authors of papers. Special thanks go to those authors of communications who managed to stay within the seven page limit and made the life easier for me. I would also like to gratefully acknowledge the support of all the above mentioned cooperating institutions. Last but not least I want to thank Springer-Verlag for excellent co-operation in publication of this volume.

Bratislava, May 1990

Branislav Rovan

CONTENTS

Invited Lectures

Communications

A LOGICAL OPERATIONAL SEMANTICS OF FULL PROLOG

Part II. Built-in Predicates for Database Manipulations

Egon Börger
IBM Germany, Heidelberg Scientific Center
Institute for Knowledge Based Systems
Tiergartenstr. 15, P.O. Box 10 30 68
D-6900 Heidelberg 1
Federal Republic of Germany

on sabbatical from: Dipartimento di Informatica
Università di Pisa
Cso Italia 40, I-56 100 PISA

Abstract

Y. Gurevich recently proposed a framework for semantics of programming concepts which directly reflects the *dynamic* and *resource-bounded* aspects of computation. This approach is based on (essentially first-order) structures that evolve over time and are finite in the same way as real computers are (so-called "dynamic algebras"). See Gurevich 1988 for the idea of dynamic algebras and its application to an operational semantics for Modula 2 (Gurevich & Morris 1988), Smalltalk (Blakley 1990), Occam (Gurevich & Moss 1990).

We use dynamic algebras to give an *operational* semantics for Prolog which, far from being hopelessly complicated, unnatural or machine-dependent, is *simple*, *natural* and *abstract* and in particular supports the process oriented understanding of programs by programmers. In spite of its abstractness, our semantics can easily be made machine executable (see Kappel 1990 for an implementation). It is designed for extensibility and as a result of the inherent extensibility of dynamic algebra semantics, we are able to proceed by stepwise refinement.

We give this semantics for the full language of Prolog including all the usual non-logical built-in predicates. Our specific aim is to provide a mathematically precise but simple framework in which *standards* can be defined rigorously and concisely and in which different implementations may be compared and judged.

Part I deals with the core of Prolog which governs the selection mechanism of clauses for goal satisfaction including backtracking and cut and closely related built-in control predicates. In the present part II the database built-in predicates are treated. Part III deals with the remaining built-in predicates.

The paper is organized as follows: in section 1 we will define the notion of dynamic algebra adapted from Gurevich 1988 and fix our language. In sections 2 and 3 we will review the core of Prolog algebras which governs the selection mechanism of clauses under backtracking and the usual control features. To take into account reactions and suggestions of practitioners to our treatment in Part I we

a) abandon the hypothesis—made there on logical grounds—that unification has to be done with occur check,

b) base our treatment on pure copying instead of interweaving copying with structure sharing, and

c) describe in just one step all the current backtracking alternatives of the goal under consideration (instead of keeping track of the first alternative which then kept track of the second, etc.)

In this way, this paper can be read independently of Part I although for sake of brevity we do not repeat here many verbal explanations, examples and motivating discussion, for which the interested reader is refered to Part I. In section 4 we describe the extensions needed for the usual built-in dabase predicates. In section 5 we will give references to related work in the literature.

1. Dynamic Algebras

The basic idea of the operational approach to semantics is to give the semantics of a programming language by an *abstract machine* for the execution of the commands of the language. Following Gurevich 1988 we consider an abstract machine to be a finite *mathematical structure*—which embodies all of the basic intuitions possessed by users of the language—together with a set of *transition rules* which reflect the execution of language commands. For the purpose of a semantics of Prolog it is sufficient to consider only first-order structures. The structures are many-sorted and partial. The latter means that the universes may be empty (at given moments in time) and that all functions are partial.

For the sake of simplicity we assume that the universe BOOL = {0,1} of Boolean values is always present. This allows the consideration of functions only, predicates being represented by their characteristic functions.

The **transition rules** we need are all of the form

IF $b(x)$ THEN $U_1(x)$

$\quad\vdots$

$U_k(x)$

where $b(x)$ is a Boolean expression of the signature of the algebra under consideration, k is a natural number and $U_1(x), \ldots, U_k(x)$ are **updates** of the two types defined below. x is a set of variables ranging over some of the universes of the algebra under consideration. If x occurs in a rule then this rule is treated as a schema (i.e. it can be applied with concrete elements of the relevant universes in place of the variables). To execute such a transition rule means to perform simultaneously all the updates $U_1(x), \ldots, U_k(x)$ in the given algebra if $b(x)$ is true. As a result of such a rule application the given algebra is "updated" to another algebra (of the same signature). For notational convenience we will use nestings of if-then-else.

As update U we allow function updates and extensions of universes. A **function update** is an equation of the form

$$f(t_1, \ldots, t_n) := t$$

where $t_1, \ldots, t_n, t$ are terms and f a function of the signature under consideration. To execute such a function update in a given algebra A means to evaluate the terms $t_1, \ldots, t_n, t$ in A, say with resulting elements $e_1, \ldots, e_n, e$ of some of the universes of A, and then to assign e as the new value of f in the argument combination $(e_1, \ldots, e_n)$. As a result we obtain a new Algebra A' of the same signature as A.