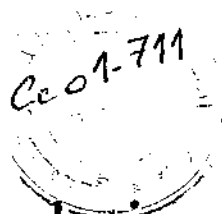


Andrzej M. Borzyszkowski
Stefan Sokołowski (Eds.)



Mathematical Foundations of Computer Science 1993

18th International Symposium, MFCS'93
Gdańsk, Poland, August 30-September 3, 1993
Proceedings



Springer-Verlag

Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editors

Gerhard Goos
 Universität Karlsruhe
 Postfach 69 80
 Vincenz-Priessnitz-Straße 1
 D-76131 Karlsruhe, Germany

Juris Hartmanis
 Cornell University
 Department of Computer Science
 4130 Upson Hall
 Ithaca, NY 14853, USA

Volume Editors

Andrzej M. Borzyszkowski
 Stefan Sokołowski
 Institute of Computer Science, Polish Academy of Sciences
 ul. Jaskowa Dolina 31, 80-252 Gdańsk, Poland

CR Subject Classification (1991): D.3.3, F.1.1, F.2.2, F.3.1, F.4.1, F.4.3, D.2.1-2, G.2.2

ISBN 3-540-57182-5 Springer-Verlag Berlin Heidelberg New York
 ISBN 0-387-57182-5 Springer-Verlag New York Berlin Heidelberg

63/8

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1993
 Printed in Germany

Typesetting: Camera-ready by authors
 Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
 45/3140-543210 - Printed on acid-free paper

Preface

This volume contains the proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science, MFCS'93, held in Gdańsk, Poland, August 30–September 3, 1993.

The MFCS symposia, organized annually in Poland and the former Czechoslovakia since 1972, have a long and well-established tradition. Over the years they have served as a meeting ground for specialists from all branches of theoretical computer science, in particular

algorithms and complexity, automata theory and theory of languages,
concurrent, distributed and real-time systems,
the theory of functional, logic and object-oriented programming,
lambda calculus and type theory,
semantics and logics of programs,

and others.

The interest in MFCS symposia seems to be growing. MFCS'93 sets a new record on the number of submissions: 133. Out of them, the Programme Committee has selected 56 for presentation, strictly on the basis of their scientific merit and appropriateness for the symposium. The committee was assisted by over 210 referees. The scientific programme includes also lectures by 12 distinguished scholars.

The proceedings contain both the invited lectures and the contributions. Unfortunately, most of the latter had to be abridged due to space limitations.

We would like to thank, first of all, the authors of the papers submitted—their interest justifies the organization of the symposia. Thanks are also due to the members of the Programme Committee and to the referees for their work put into the evaluation of the papers. Finally, we thank Springer-Verlag for a long-term cooperation with MFCS symposia.

A number of institutions have financially contributed to the success of the symposium. We gratefully acknowledge the support from Stefan Batory Foundation, *Gdańsk Development Method* research project, Warsaw University, Wrocław University, Gdańsk University.

MFCS'93 is organized by the Institute of Computer Science of the Polish Academy of Sciences. The Organizing Committee consists of Marek A. Bednarczyk (chairman), Andrzej M. Borzyszkowski, Ryszard Kubiak, Wiesław Pawłowski and Stefan Sokółowski.

Gdańsk, June 1993

Andrzej M. Borzyszkowski
Stefan Sokółowski

Programme Committee

J. Adámek (Praha)	E. Moggi (Genova)
M. A. Bednarczyk (Gdańsk)	V. Nepomniaschy (Novosibirsk)
J. A. Bergstra (Amsterdam)	E.-R. Olderog (Oldenburg)
A. M. Borzyszkowski (Gdańsk)	F. Orcjas (Barcelona)
B. Courcelle (Bordeaux)	A. Poigné (St. Augustin)
H.-D. Ehrich (Braunschweig)	I. Sain (Budapest)
I. Havel (Praha)	D. A. Schmidt (Manhattan, Kansas)
G. Huet (Paris)	S. Sokolowski (Gdańsk), chair
N. Jones (København)	M. Wirsing (München)

Non-PC Referees

K. Ambos-Spies	T. Coquand	M. Grosse-Rhode	C. Kenyon
N. Andersen	G. Costa	R. Grosu	D. Kesner
H. R. Andersen	L. Csirmaz	C. A. Gunter	N. Klarlund
H. Andréka	P.-L. Curien	P. Hajnal	J. W. Klop
M. Anselmo	L. Czaja	M. R. Hansen	S. Klusener
P. Audebaud	R. De Nicola	T. Hardin	B. Konikowska
J. L. Balcazar	P. Dembiński	J. Hatcliff	S. Kotov
A. Banerjee	M. Dezani	M. Hennessy	V. Koubek
P. Barth	R. Diaconescu	M. Hermann	H.-J. Kreowski
M. Bauderon	J. Diaz	R. Herzig	B. Krieg-Brückner
T. Bétréma	V. Diekert	L. Holenderski	M. Krivánek
U. Berger	G. Dowek	R. Howell	F. Kröger
J. Berstel	D. Dranidis	J. Hromkovič	R. Kubiak
J. Biskup	P. Dybjer	M. Huhn	A. Kučera
A. J. Blikle	H. Ehrig	H. Hungar	G. Kucherov
J. Bradfield	V. Evstigneev	G. Jarzembksi	Á. Kurucz
F. J. Brandenburg	G.-L. Ferrari	J. Jędrzejowicz	H. Le Verge
R. Breu	J. L. Fiadeiro	B. Jenner	U. Lechner
C. Brown	H. Fleischhack	C. T. Jensen	C. Lengauer
F. Broy	W. J. Fokink	J. Jørgensen	P. Lincoln
A. Bystrov	G. S. Frandsen	B. Josko	U. W. Lipeck
R. Casas	D. de Frutos-Escrig	J.-P. Jouannaud	I. Litovsky
I. Castellani	J. Gabarro	A. Jung	G. Longo
M. V. Cengarle	A. Gammelgaard	S. Kaes	A. Lozano
M. M. Chiara	G. Ghelli	G. Kahn	Z. Luo
C. Choffrut	P. Y. Gloess	R. Kaivola	T. Maibaum
J. Chomicki	L. Godo	J. Kari	K. Malmkjær
J. Clausen	M. Gogolla	S. Kars	V. Manca
L. Colson	J. Goguen	J. Katajainen	S. Martini

E. W. Mayr	C. Palamidessi	M. Rusinowitch	A. Sulimov
N. D. N. Measor	C. Paulin	G. Saake	A. Szalas
S. Meiser	W. Pawlowski	N. Saheb	A. Szepietowski
V. Ménéssier-Morain	W. Penczek	A. Salomaa	A. Tarlecki
A. Middeldorp	H. Peterreins	D. Sands	E. Teniente
S. Mikulás	W. Phoa	M. Schenke	J. Toran
P. B. Miltersen	B. Pierce	G. Schreiber	J. L. Träff
T. Mogensen	G. M. Pinna	M. Schwartzbach	E. Upfal
B. Monien	D. Pisinger	A. Schwill	P. Urzyczyn
L. Moss	A. Pitts	G. Scollo	Y. Venema
P. D. Mosses	V. Pollara	G. Serény	R. Verbeek
A. W. Mostowski	J. Power	M. Serna	A. Vinacua
F. Mráz	T. Przytycka	H. Shi	I. B. Virbitskaite
A. Mück	Z. Qian	N. V. Shilov	S. Vorobyov
D. Murphy	A. Raspand	A. Simon	D. Wätjen
I. Németi	G. Rauzy	G. Snelting	K. Weihrauch
M. Nielsen	G. Reggio	S. Soloviev	B. Werner
R. Nieuwenhuis	J. Rehof	E. Sopena	J. Winkowski
U. Nilsson	W. Reisig	Z. Sławski	P. Winter
E. Ohlebusch	B. Reus	M. Srebrny	M. Zaionc
L. Ong	J. Roman	R. Statman	E. Zucca
E. Orłowska	M. Rosendahl	O. Stepankova	A. Zvonkin
Y. Ortega-Mallén	G. Rosolini	T. Streicher	J. Zwiers
L. Pacholski	B. Rován	W. Struckmann	
P. Pačzkowski	M. Rodríguez-Artalejo		

A couple of referees have remained anonymous to the editors of the volume. We are as thankful to them as to the ones listed above for their work.

Table of Contents

Invited lectures

Krzysztof R. Apt:

On the unification free Prolog programs

by K. R. Apt and S. Etalle 1

André Arnold:

Equivalences and preorders of transition systems

by A. Arnold and A. Dicky 20

Rod Burstall:

Deliverables: a categorical approach to program development in type theory

by J. McKinna and R. Burstall 32

Volker Diekert:

Complex and complex-like traces 68

Matthew Hennessy:

Symbolic bisimulations (abstract) 83

Furio Honsell:

Some results on the full abstraction problem for restricted lambda calculi

by F. Honsell and M. Lenisa 84

Robin Milner:

Action calculi, or syntactic action structures 105

Andrew M. Pitts:

Observable properties of higher order functions that dynamically create local names, or: What's new?

by A. M. Pitts and I. D. B. Stark 122

Vaughan Pratt:

The second calculus of binary relations 142

Grzegorz Rozenberg:

An introduction to dynamic labeled 2-structures

by A. Ehrenfeucht and G. Rozenberg 156

Arto K. Salomaa:

Post correspondence problem: Primitivity and interrelations with complexity classes

by A. Mateescu and A. Salomaa 174

Philip Wadler:

A taste of linear logic 185

Contributions

L. Alonso and R. Schott	
<i>On the tree inclusion problem</i>	211
R. Arnadio	
<i>On the adequacy of per models</i>	222
V. Arvind, J. Köbler, and M. Mundhenk	
<i>Hausdorff reductions to sparse sets and to sets of high information content</i>	232
E. Astesiano, G. Reggio, and E. Zucca	
<i>Stores as homomorphisms and their transformations</i>	242
J. W. de Bakker, F. van Breugel, and A. de Bruin	
<i>Comparative semantics for linear arrays of communicating processes, a study of the UNIX fork and pipe commands</i>	252
D. Beauquier and A. Podelski	
<i>Rabin tree automata and finite monoids</i>	262
M. Benke	
<i>Efficient type reconstruction in the presence of inheritance</i>	272
J. Berstel and P. Séébold	
<i>A characterization of Sturmian morphisms</i>	281
H. L. Bodlaender and K. Jansen	
<i>On the complexity of scheduling incompatible jobs with unit-times</i>	291
M. Bonsangue and J. N. Kok	
<i>Isomorphisms between predicate and state transformers</i>	301
Liming Cai and Jianer Chen	
<i>On the amount of nondeterminism and the power of verifying</i>	311
I. Castellani	
<i>Observing distribution in processes</i>	321
C. Cérin and A. Petit	
<i>Speedup of recognizable trace languages</i>	332
M. Cerioli and J. Meseguer	
<i>May I borrow your logic?</i>	342
S. Chaudhuri, T. Hagerup, and R. Raman	
<i>Approximate and exact deterministic parallel selection</i>	352
Jian Chen and J. Staples	
<i>Defining soft sortedness by abstract interpretation</i>	362

Liang Chen	
<i>A model for real-time process algebras</i>	372
J. F. Costa, A. Sernadas, and C. Sernadas	
<i>Data encapsulation and modularity: Three views of inheritance</i>	382
K. Culik II and J. Kari	
<i>Image compression using weighted finite automata</i>	392
M. Dezani-Ciancaglini, U. de'Liguoro, and A. Piperno	
<i>Filter models for a parallel and non deterministic λ-calculus</i>	403
P. Di Gianantonio	
<i>Real number computability and domain theory</i>	413
Z. B. Diskin and I. D. Beylin	
<i>Lambda substitution algebras</i>	423
B. Durand	
<i>Global properties of 2D cellular automata: Some complexity results</i>	433
U. Engberg and G. Winskel	
<i>Completeness results for linear logic on Petri nets</i>	442
A. Fantechi, S. Gnesi, and V. Peticaroli	
<i>An expressive logic for basic process algebra</i>	453
G. S. Frandsen, P. B. Miltersen, and S. Skyum	
<i>The complexity of finding replicas using equality tests</i>	463
R. J. van Glabbeek	
<i>A complete axiomatization for branching bisimulation congruence of finite-state behaviours</i>	473
E. de Haas and P. van Emde Boas	
<i>Object oriented application flow graphs and their semantics</i>	485
J. Hromkovič, J. Kari, and L. Kari	
<i>Some hierarchies for the communication complexity measures of cooperating grammar systems</i>	495
L. Ibarra and D. Richards	
<i>Efficient parallel graph algorithms based on open ear decomposition</i>	506
O. H. Ibarra and N. Trân	
<i>On the communication complexity of parallel computation</i>	517
P. Jančar, F. Mráz, and M. Plátek	
<i>A taxonomy of forgetting automata</i>	527

He Jifeng	
<i>Hybrid parallel programming and implementation of synchronised communication</i>	537
A. Kiehn	
<i>Proof systems for cause based equivalences</i>	547
B. Martin	
<i>A uniform universal CREW PRAM</i>	557
D. Murphy	
<i>Observing located concurrency</i>	566
P. Narbel	
<i>The boundary of substitution systems</i>	577
J. Neraud	
<i>New algorithms for detecting morphic images of a word</i>	588
P. P/aczkowski	
<i>Ignoring nonessential interleavings in assertional reasoning on concurrent programs</i>	598
M. Parigot and P. Rozière	
<i>Constant time reductions in λ-calculus</i>	608
F. Parisi-Presicce and S. Veghioni	
<i>Heterogeneous unified algebra</i>	618
D. Pigozzi and A. Salibra	
<i>A representation theorem for lambda abstraction algebras</i>	629
R. Pliuškevičius	
<i>On saturated calculi for a linear temporal logic</i>	640
H. Puhlmann	
<i>The snack powerdomain for database semantics</i>	650
B. Reus and T. Streicher	
<i>Verifying properties of module construction in type theory</i>	660
P. Ružička and J. Waczulík	
<i>On time-space trade-offs in dynamic graph pebbling</i>	671
V. Sassone, M. Nielsen, and G. Winskel	
<i>Deterministic behavioural models for concurrency</i>	682
D. Scholefield, H. Zedan, and He Jifeng	
<i>Real-time refinement: Semantics and application</i>	693
B. Steffen and C. Weise	
<i>Deciding testing equivalence for real-time processes with dense time</i>	703

W. Stephan and A. Wolpers	
<i>A calculus for higher order procedures with global variables</i>	714
D. Stevens	
<i>Variable substitution with iconic combinators</i>	724
R. Treinen	
<i>Feature constraints with first-class features</i>	734
D. Wagner and F. Wagner	
<i>Between Min Cut and Graph Bisection</i>	744
E. Wanke	
<i>Paths and cycles in finite periodic graphs</i>	751
Jilei Yin and Zhu Hong	
<i>Learning decision lists from noisy examples</i>	761
N. Zabel	
<i>Analytic tableaux for finite and infinite Post logics</i>	767
Subject index	777
Author index	782

On the Unification Free Prolog Programs

Krzysztof R. Apt¹ and Sandro Etalle²

¹ CWI

P.O. Box 4079, 1009 AB Amsterdam, The Netherlands
and

Faculty of Mathematics and Computer Science
University of Amsterdam, Plantage Muidergracht 24
1018 TV Amsterdam, The Netherlands

² Dipartimento di Matematica Pura ed Applicata
Università di Padova

Via Belzoni 7, 35131 Padova, Italy

Abstract. We provide simple conditions which allow us to conclude that in case of several well-known Prolog programs the unification algorithm can be replaced by iterated matching. The main tools used here are types and generic expressions for types. As already noticed by other researchers, such a replacement offers a possibility of improving the efficiency of program's execution.

Notes. The work of the first author was partly supported by ESPRIT Basic Research Action 6810 (Compulog 2). This research was done partly during the second author's stay at Centre for Mathematics and Computer Science, Amsterdam.

1 Introduction

Unification is heralded as one of the crucial features offered by Prolog, so it is natural to ask whether it is actually used in specific programs. The aim of this paper is to identify natural conditions under which unification can be replaced by iterated matching and to show that they are applicable to several well-known Prolog programs. These conditions can be statically checked without analyzing the search trees for the queries. For programs which use ground inputs they can be efficiently tested.

The problem of replacing unification by iterated matching was already studied in the literature by a number of researchers – see e.g. Deransart and Maluszynski [DM85b], Maluszynski and Komorowski [MK85] and Attali and Franchi-Zannettacci [AFZ88]. As in the previous works on this subject, we use modes, which indicate how the arguments of a relation should be used. Our results improve upon the previous ones due to the additional use of types. This allows us to deal with non-ground inputs.

We use here a simple notion of a type, which is a set of terms closed under substitution. The main tool in our approach is the concept of a *generic expression*. Intuitively, a term s is a generic expression for a type T if it is more general than all elements of T which unify with s . This simple notion turns out to be crucial here,

because surprisingly often the input positions of the heads of program clauses are filled in by generic expressions for appropriate types.

We combine in our analysis the use of generic expressions with the notion of a *well-typed program*, recently introduced by Bronsard, Lakshman and Reddy [BLR92], which allows us to ensure that the input positions of the selected atoms remain correctly typed. As the table included at the end of this paper shows, our results can be applied to astonishingly many Prolog programs.

2 Preliminaries

In what follows we study logic programs executed by means of the *LD-resolution*, which consists of the SLD-resolution combined with the leftmost selection rule. An SLD-derivation in which the leftmost selection rule is used is called an *LD-derivation*. We allow in programs various first-order built-in's, like $=$, $\neq$, $>$, etc, and assume that they are resolved in the way conforming to their interpretation.

We work here with *queries*, that is sequences of atoms, instead of *goals*, that is constructs of the form $\leftarrow Q$, where Q is a query. Apart from this we use the standard notation of Lloyd [Llo87] and Apt [Apt90]. In particular, given a syntactic construct E (so for example, a term, an atom or a set of equations) we denote by $\text{Var}(E)$ the set of the variables appearing in E . Given a substitution $\theta = \{x_1/t_1, \dots, x_n/t_n\}$ we denote by $\text{Dom}(\theta)$ the set of variables $\{x_1, \dots, x_n\}$, by $\text{Range}(\theta)$ the set of terms $\{t_1, \dots, t_n\}$, and by $\text{Ran}(\theta)$ the set of variables appearing in $\{t_1, \dots, t_n\}$. Finally, we define $\text{Var}(\theta) = \text{Dom}(\theta) \cup \text{Ran}(\theta)$.

Recall that a substitution θ is called *grounding* if $\text{Ran}(\theta)$ is empty, and is called a *renaming* if it is a permutation of the variables in $\text{Dom}(\theta)$. Given a substitution θ and a set of variables V , we denote by $\theta|V$ the substitution obtained from θ by restricting its domain to V .

2.1 Unifiers

Given two sequences of terms $s = s_1, \dots, s_n$ and $t = t_1, \dots, t_n$ of the same length we abbreviate the set of equations $\{s_1 = t_1, \dots, s_n = t_n\}$ to $\{s = t\}$ and the sequence $s_1\theta, \dots, s_n\theta$ to $s\theta$. Two atoms can unify only if they have the same relation symbol. With two atoms $p(s)$ and $p(t)$ to be unified we associate the set of equations $\{s = t\}$. In the applications we often refer to this set as $p(s) = p(t)$. A substitution θ such that $s\theta = t\theta$ is called a *unifier* of the set of equations $\{s = t\}$. Thus the set of equations $\{s = t\}$ has the same unifiers as the atoms $p(s)$ and $p(t)$.

A unifier θ of a set of equations E is called a *most general unifier* (in short *mgu*) of E if it is more general than all unifiers of E . An mgu θ of a set of equations E is called *relevant* if $\text{Var}(\theta) \subseteq \text{Var}(E)$.

The following lemma was proved in Lassez, Marriot and Maher [LMM88].

Lemma 1. *Let θ_1 and θ_2 be mgu's of a set of equations. Then for some renaming η we have $\theta_2 = \theta_1\eta$.* $\square$

Finally, the following well-known lemma allows us to search for mgu's in an iterative fashion.