Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

431

A. Arnold (Ed.)

CAAP '90

15th Colloquium on Trees in Algebra and Programming
Copenhagen, Denmark, May 15–18, 1990
Proceedings

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

Editor
André Arnold
Université de Bordeaux I, Laboratoire d'Informatique
351, Cours de la Libération, F-33405 Talence, France

CR Subject Classification (1987): D.1, E.1, F

ISBN 3-540-52590-4 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-52590-4 Springer-Verlag New York Berlin Heidelberg

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

Preface

This volume contains the Proceedings of the Fifteenth Colloquium on Trees in Algebra and Programming (CAAP '90), held May 15-18, 1990, in Copenhagen.

The preceding fourteen colloquia were held in France (the first five in Lille), Italy, Germany and Spain. In 1985 CAAP was integrated with TAPSOFT, which takes place every second year (Berlin in 1985, Pisa in 1987, Barcelona in 1989). This integration is a way to keep theoretical research close to practical developments in computer science. Another step in this direction was to hold CAAP jointly with ESOP, the European Symposium on Programming in 1988, in Nancy. In 1990, CAAP is again held jointly with ESOP. The Proceedings of ESOP '90 are published in another LNCS volume, a twin to this one.

At first the colloquium series was devoted to the algebraic and combinatorial properties of trees, and their role in various fields of computer science. Nowadays trees are as well established in computer science as strings, but many other discrete structures, graphs for instance, are also being used. Therefore the scope of CAAP has been extended to any kind of discrete structures (strings, trees, graphs, ...), their logical algebraic and combinatorial properties, and their applications in computer science: syntax and semantics of programming languages, design and analysis of algorithms, etc.

Forty-six papers were submitted, covering almost all the topics mentioned in the call for papers. Sixteen were selected by the Program Committee. The choice was made on the quality of the papers, but it turns out that they are a representative sample of the submitted ones. On the average the submitted papers were rather good and the task of the Program Committee was not always very easy. In particular several papers contained new and interesting ideas, but were not worked out well enough.

The two invited lecturers are X.G. Viennot and D. Harel. I am pleased to thank these two well-recognized scholars for accepting the invitation of the Program Committee.

I should like to thank the Program Committee members and all the referees for the work they did and their help in preparing the program.

I should like to thank Neil Jones, Chairman of the ESOP Program Committee, and Nils Andersen, Chairman of Local Arrangements, for the organisation of this joint conference, and thus of this fifteenth CAAP.

A. Arnold

Program Committee

André Arnold (Bordeaux, Chairman)

Giorgio Ausiello (Rome)
Franz J. Brandenburg (Passau)
Bruno Courcelle (Bordeaux)
Max Dauchet (Lille)
Josep Diaz (Barcelona)
Hartmut Ehrig (Berlin West)
Joost Engelfriet (Leiden)

Gilberto Filé (Padova)
David Harel (Rehovot)
Matthias Jantzen (Hamburg)
Giuseppe Longo (Pisa)
Maurice Nivat (Paris)
Sven Skyum (Århus)
Pierre Wolper (Liège)

The program of CAAP'90 offered two invited talks which are included in this volume. The Program Committee thanks the invited lecturers:

David Harel (The Weizmann Institute of Science, Rehovot, Israel)
Xavier Gérard Viennot (University of Bordeaux I, France)

List of Referees for CAAP '90

S. Abiteboul
S. Abramsky
R. Amadio
P. America
P.R.J. Asveld
E. Badouel
A. Behboud
H. Bodlaender
A. Bossi
F. Bossut
G. Boudol
H. Carstensen
I. Castellani
P. Casteran
L. Colussi
H. Comon
P. Darondeau
J.P. Delahaye
V. Diekert
U. Engberg
J. Gabarro
B. Gamatié
G. Gambosi
R. Gilleron

A. Habel
D. Hauschildt
D. Hofbauer
H.J. Hoogeboom
J. Karhumaki
B. Kirsig
J.W. Klop
H.J. Kreowski
Y. Lafont
M. Latteux
G. Leha
M. Lenzerini
D. Lepore
J.L. Loquidé
M. Löwe
B. Mahr
O. Maler
C. Marche
B.H. Mayoh
A. Mazurkiewicz
Y. Métivier
U. Nanni
P. Naudin
H.R. Nielson

P. Padawitz
F. Parisi-Presicce
D. Peleg
J.G. Penaud
M. Protasi
L. Puel
J.C. Raoult
E.M. Schmidt
G. Senizergues
J. Skurczynski
L. Staiger
B. Steffen
M. Steinby
R. Strandh
W. Thomas
S. Tison
E. Upfal
S. Valentini
P. Van Emde Boas
H. Vogler
W. Vogler
E. Welzl
G. Winskel
D. Wolz
S. Yoccoz

Table of Contents

On the Power of Bounded Concurrency II: The Pushdown Automata Level [†]

Tirza Hirst

Dept. of Mathematics & Computer Science
Bar-Ilan University, Ramat Gan, Israel

and

David Harel[‡]

Dept. of Applied Mathematics & Computer Science
The Weizmann Institute of Science, Rehovot 76100, Israel

Abstract.

This is the second in a series of papers on the inherent power of bounded cooperative concurrency, whereby an automaton can be in some bounded number of states that cooperate in accepting the input. In this paper we deal with the level of pushdown automata. We are interested in differencies in power of expression and in discrepencies in succinctness between variants of pda's that incorporate nondeterminism, pure parallelism and bounded cooperative concurrency. In particular, our results provide further evidence for the general observation that the latter feature provides inherent exponential power, in both upper and lower bound senses, regardless of whether or not the two former features are also present. While we use the language of statecharts to capture these features, our results are extremely robust, and hold also for bounded versions of virtually all other concurrent languages.

1. Introduction

Classical models of computation, such as Turing machines and various kinds of automata, have been enriched with existential and universal branching to capture parallelism. However, unlike the constructs used in the study of real distributed processes and protocols, in these types of branching no cooperation takes place between the spawned processes, except when time comes to decide whether the input should be accepted. In Turing machines and pushdown automata, for example, this fact manifests itself in the totally separate tapes or pushdown stacks that are assumed to be generated whenever branching (of either kind) takes place. Thus, branching essentially produces separate computations, the results of which

[†] This paper is based on part of the M.Sc. thesis of the first-listed author [Hi], supervised by the second-listed author.

[‡] This author's research was supported in part by a grant from the Gutwirth Foundation.