Lecture Notes in
Computer Science

Edited by G. Goos and J. Hartmanis

249

TAPSOFT '87

TAP SOFT '87

Proceedings of the International Joint Conference on Theory and Practice of Software Development
Pisa, Italy, March 23–27, 1987

Volume 1:
Advanced Seminar on Foundations of Innovative Software Development I and Colloquium on Trees in Algebra and Programming (CAAP '87)

Edited by Hartmut Ehrig, Robert Kowalski, Giorgio Levi and Ugo Montanari

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo

Editors

Hartmut Ehrig
Technische Universität Berlin
Fachbereich 20, Informatik, SWT FR5-6
Franklinstr. 28/29, D-1000 Berlin 10

Robert Kowalski
Imperial College of Science and Technology
180 Queen's Gate, London SW7 2BZ, England

Giorgio Levi
Ugo Montanari
Dipartimento di Informatica, Università di Pisa
Corso Italia 40, I-56100 Pisa

CR Subject Classification (1987): F.0-4

ISBN 3-540-17660-8 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-17660-8 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210

PREFACE

TAPSOFT '87 is the Second International Joint Conference on Theory and Practice of Software Development.

TAPSOFT '87 is being held from March 23 to March 27, 1987 in Pisa. TAPSOFT '87 has been organized by Dipartimento di Informatica (Università di Pisa), I.E.I. - C.N.R. and CNUCE - C.N.R., and has been supported by AICA and EATCS.

TAPSOFT '87 consists of three parts:

Advanced Seminar on Foundations of Innovative Software Development

New directions in software development have been proposed, on the basis of recent technological and theoretical advances. Following these trends, the software production process should be made more rigorous, and its result should be expressed in a more abstract and understandable form.
The aim of the Advanced Seminar is to bring together leading experts in the various fields which form the foundations of this renovation still in progress and to provide a forum to discuss the possible integration of available theories and methods in view of their applications.
The Advanced Seminar will consist of a number of invited talks, two panel discussions and several working groups. The invited talks will be either long, i.e. comprehensive and general, or short, i.e. dedicated to hot topics.

Invited Speakers

E. Astesiano (Univ. Genova)
K. Clark (Imp. C., London)
K. Furukawa (ICOT, Tokyo)
J. Goguen (SRI, Menlo Park)
G. Huet (INRIA, Paris)
R. Milner (Univ.Edinburgh)
M. Nivat (LITP, Paris)
J. Thatcher (IBM, Yorktown Heights)
D. Warren (Univ. Manchester)

Panels

• On Industrial Activity and Trends. Chairman: J. Goguen (SRI, Menlo Park)
• The Future of Software Engineering. Chairman: D. Bjørner (Lyngby)

The seminar organizers are H. Ehrig (Tech. Univ. Berlin) G. Levi (Univ. Pisa)
R. Kowalski (Imperial College, London) U. Montanari (Univ. Pisa)

Colloquium on Trees in Algebra and Programming

Traditionally, the topics of the Colloquium cover a wider area of theoretical Computer Science than that indicated by the title. Actually, topics include the formal aspects and properties of trees and, more generally, of combinatorial and algebraic structures in all fields of Computer Science.
Besides the customary topics, in keeping with the overall theme of TAPSOFT, the program will include contributions related to specifications, communicating systems and type theory.

The preceding eleven colloquia were held in France and Italy as autonomous conferences, except in Berlin 1985, when for the first time CAAP was integrated into the TAPSOFT Conference.

In keeping with the tradition of CAAP as well as with the overall theme of the TAPSOFT conference, the selected papers are presented in the sections listed below.

- Algorithms
- Proving techniques
- Algebraic specifications
- Concurrency
- Foundations

The program committee for CAAP '87 is the following:

A. Arnold, Bordeaux
J. de Bakker, Amsterdam
B. Buchberger, Linz
J. Diaz, Barcelona
Ph. Flajolet, Paris
H. Ganzinger, Dortmund
P. Mosses, Aarhus
J. Thatcher, Yorktown Heights
M. Wirsing, Passau
G. Ausiello, Roma
A. Bertoni, Milano
M. Dauchet, Lille
H. Ehrig, Berlin
N. Francez, Haifa
U. Montanari, Pisa (Chairman)
M. Nivat, Paris
G. Winskel, Cambridge

Colloquium on Functional and Logic Programming and Specifications

In keeping with the overall theme of the TAPSOFT conferences, CFPL focuses on those aspects of Functional and Logic Programming which are most important in innovative software development. The integration of formal methods and practical aspects of software production is also stressed.

The selected papers are presented in six sessions covering the following topics.

- Theory and Semantics of Functional Languages
- Types, Polymorphism and Abstract Data Type Specifications
- Unification of Functional and Logic Programming Languages
- Program Proving and Transformation
- Language Features and Compilation in Logic Programming
- Implementation Techniques

The Programme Committee for CFLP is the following

C. Böhm, Roma
K. Clark, London
K. Furukawa, Tokyo
H. Gallaire, München
C. Ghezzi, Milano
J. Goguen, Menlo Park
G. Huet, Paris
G. Kahn, Sophia Antipolis
R. Kowalski, London
G. Levi, Pisa (Chairman)
B. Mahr, Berlin
A. Martelli, Torino
R. Milner, Edinburgh
L. Moniz Pereira, Lisboa
E. Sandewall, Linköping
E. Shapiro, Rehovot
D. Warren, Manchester

The TAPSOFT '87 Conference proceedings are published in advance of the conference in two volumes. The first volume includes the final versions of 17 papers from CAAP '87, selected from a total of 51 submitted papers. The second volume includes the final versions of 17 papers from CFLP, selected from a total of 80 submitted papers. Invited papers from the Advanced Seminar are divided between the two volumes.

We would like to extend our sincere thanks to all the Program Committee members as well as to the referees listed below for their care in reviewing and selecting the submitted papers:
J. Alegria, A. Alfons, S. Anderson, J.L. Balcázar, F. Barbic, R. Barbuti, M. Bellia, R. Bird, E. Börger, P.G. Bosco, A. Bossi, G. Boudol, K. Broda, D. Brough, D. Chan, L. Carlucci Aiello, G. Castelli, T. Chikayama, T. Chusho, E. Ciapessoni, N. Cocco, L. Colussi, M. Coppo, T. Coquand, B. Courcelle, G. Cousineau, W. Coy, P.L. Curien, A. Davison, P. Degano, R. De Nicola, M. Dezani, M. Dincbas, M. Ducassé, P. Dufresne, J. Ebert, B. Eggers, P. van Emde Boas, R. Enders, G. Engels, K. Estenfeld, E. Fachini, A. Fantechi, I. Foster, D. Frutos, J. Gabarro, D. Gabbay, F. Galdbay, G. Gambosi, G. Ghelli, P. Giannini, M. Goldwurm, A. Goto, S. Goto, G. Guida, C. Gunter, T. Iato, H. Habel, M. Hagiya, N. Halbwacks, H. Hansen, S. Haqqlund, J. Heering, P. Henderson, R. Hennincker, D. Henry de Villeneuve, C. Hogger, F. Honsell, M. Huntback, H. Hussmann, P. Inverardi, R.C.L. Koymans, L. Kott, H.J. Kreowski, F. Kriwaczek, S. Kunifuji, Y. Lafont, B. Lang, R. Lasas, A.

Laville, P. Le Cheradec, K. Leeb, B. Lennartsson, J.J. Levy, M. Lindqvist, A. Llamosi, G. Lolli, G. Longo, J.A. Makowski, V. Manca, P. Mancarella, D. Mandrioli, M. Manny, A. Marchetti Spaccamela, I. Margaria, M. Martelli, L. Mascoet, Y. Matsumoto, G. Mauri, B.H. Mayoh, F. McCabe, J. Meseguer, J.J.Ch. Meyer, C. Moiso, B. Möller, C. Montangero, K. Moody, A. Mycroft, F. Nickl, M. Nielsen, F. Nielson, F. Nürnberg, M.E. Occhiuto, F.J. Oles, F. Orejas, M. Ornaghi, R. Orsini, P. Padawitz, C. Palamidessi, D. Pedreschi, P. Pepper, A. Pettorossi, A. Poigné, A. Porto, M. Protasi, G. Ringwood, J. Roman, S. Ronchi Della Rocca, G. Rossi, I. Kott, T. Sakurai,D. Sannella, D. Sartini, T. Sato, R. Schuster, M. Sergot, D. Siefkes, M. Smyth, T. Streicher, A. Suarez, Y. Takayama, J. Tanaka, A. Tarlecki, W. Thomas, M. Tofte, S. Tomura, J. Toran, M. Torelli, J.V. Tucker, F. Turini, T. Yuasa, F.W. Vaandrager, B. Vauquelin, B. Venneri, M. Venturini Zilli, H. Wagener, E.G. Wagner, M. Wallace, P. Weis, M. Zacchi, B. Zimmermann

We gratefully acknowledge the financial support provided by the following Institutions and Companies:

- Comune di Pisa
- C.N.R.
  - Presidenza
  - Comitato Nazionale per l'Ingegneria
  - Comitato Nazionale per le Scienze Matematiche
  - CNUCE
  - I.E.I.
- Dipartimento di Informatica, Università di Pisa
- Elsag, Genova
- Enidata, Milano
- IBM Italia, Roma
- List, Pisa
- Olivetti, Ivrea
- Selenia, Roma
- Sipe, Roma
- Systems & Management, Torino
- Tecsiel, Roma
- Università di Pisa

We wish to express our gratitude to the members of the Local Arrangement Committee: P. Asirelli, R. Barbuti, P. Degano (Chairman), A. Fantechi, P. Mancarella, M. Martelli, F. Tarini and F. Turini. Without their help, the Conference would not have been possible.

Pisa, March 1987

Hartmut Ehrig
Institut für Software und Theoretische Informatik
Technische Universität Berlin

Robert A. Kowalski
Dept of Computing and Control
Imperial College London

Giorgio Levi
Dipartimento di Informatica
Università di Pisa

Ugo Montanari
Dipartimento di Informatica
Università di Pisa

CONTENTS OF VOLUME 1

CONTENTS OF VOLUME 2

ON THE COMPLEXITY OF BRANCHING PROGRAMS AND DECISION TREES FOR CLIQUE FUNCTIONS

Ingo Wegener*
FB 20-Informatik, Johann Wolfgang Goethe-Universität,
6000 Frankfurt a.M., Fed.Rep. of Germany

Abstract

Because of the slow progress in proving lower bounds on the circuit complexity of Boolean functions one is interested in restricted models of Boolean circuits like depth restricted circuits, decision trees, branching programs, width-k branching programs and k-times-only branching programs. We prove here exponential lower bounds on the decision tree complexity of clique functions. For one-time-only branching programs we prove for k-clique functions large polynomial lower bounds if k is fixed and exponential lower bounds for k increasing with n. Finally we introduce the hierarchy of the classes $BP_k(P)$ of all sequences of Boolean functions which may be computed by k-times-only branching programs of polynomial size. We show constructively that $BP_1(P)$ is a proper subset of $BP_2(P)$.

1. INTRODUCTION

Until now one knows only a few poor methods for the proof of lower bounds on the circuit complexity of explicitly defined Boolean functions. Therefore one has considered since a long time restricted models like formulae, monotone circuits, branching programs, contact schemes (Nechiporuk [9]) and also restricted models of branching programs like width restricted branching programs (Ajtai et al. [1], Barrington [2], Borodin/Dolev/Fich/Paul [3], Chandra/Furst/Lipton [4], Pudlák [10] and Yao [15]) and (depth restricted) k-times-only branching programs (Ajtai et al. [1], Dunne [5], Kriegel/Waack [7], Masek [8], Pudlák/Žák [11] and Wegener [12], [13], [14]). We assume that the reader is familiar with Boolean circuits and formulae. A decision tree is a directed, labelled binary tree where the inner nodes are labelled by Boolean variables and the leaves by Boolean constants. One starts at the root and after reaching an inner node one tests that variable which

* Supported in part by DFG-grants No. We 1066/1-2 and Me 872/1-1

is the label of the node. If its value is 0 (or 1) one goes to the left (or right) successor. The label of the leaf one reaches is the value of the function computed by the decision tree. DT(f), the decision tree complexity of f, is the minimal number of inner nodes of a decision tree for f. A branching program is a directed acylic graph with one source where the computation starts, inner nodes of outdegree 2 and sinks of outdegree 0. The labelling and the mode of computation is similar to that of decision trees, the proper complexity measure is BP(f). Width-k branching programs are levelled and have not more than k nodes on each level. On the other hand k-times-only branching programs (BP_ks) have a depth restriction. One is allowed to test each variable on each path of computation only for k times, the complexity measure is $BP_k(f)$. Obviously all optimal decision trees are BP_1s. A computation of a BP_k needs at most kn points of time while the computation time for width-k branching programs is of the same size as the number of nodes of the program. The problem is to decide which functions may be computed efficiently even by restricted branching programs and for which functions one can prove large lower bounds in the restricted models. For width restricted branching programs this problem has a surprising solution (Barrington [2]). Sequences of Boolean functions can be computed by branching programs of polynomial size and constant width iff they can be computed by circuits of polynomial size and logarithmic depth.

We are far from similar results for depth restricted branching programs. For the motivation of BP_ks we refer to Wegener [12]. Masek [8], Pudlák/Žák [11] and Ajtai et al. [1] proved tight relations between the size of BPs (BP_ks) and the space complexity of Turing machines (so-called eraser Turing machines).

The purpose of this paper is to present methods for the proof of lower bounds on the decision tree and BP_1 complexity of Boolean functions. We apply these methods to clique functions. The clique function f_k^n where $3 \leq k \leq n-1$ is defined on $N = \binom{n}{2}$ variables corresponding to the possible edges of an n-vertex graph. f_k^n computes 1 iff the graph specified by the variables contains a k-clique.

In Chapter 2 we consider decision trees and present some general lower bound techniques.

In Chapter 3 we present our main method, a lower bound method for BP_1s. We show which computation paths cannot lead to the same computation node in a BP_1, since a BP_1 cannot separate the situations by repeating an old test. This method leads to strong lower bounds for clique functions. The largest lower bounds we obtain are of size $\exp(\Omega(N^{1/2}))$ for the number of variables N. This method has first been presented in a preliminary version of this paper [13]. Dunne [5] applied this method to other functions. Recently Ajtai et al. [1] and Kriegel/Waack [7] used this approach to prove even lower bounds of size $\exp(\Omega(N))$.

In Chapter 4 we consider the hierarchy problem for BP_ks. Let $BP_k(P)$ be the class of sequences of Boolean functions computable by BP_ks of polynomial size. We conjecture that these classes build a proper hierarchy, i.e. $BP_{k-1}(P)$ is a proper subclass of $BP_k(P)$. We present candidates which may separate the classes and prove that $BP_1(P)$ is a proper subclass of $BP_2(P)$.

2. ON THE DECISION TREE COMPLEXITY OF CLIQUE FUNCTIONS

Similar to Boolean formulae we count here the number of leaves instead of the number of inner nodes of decision trees. This number is only by 1 larger than the decision tree complexity.

Definition 1: Let $DT^*(f):=DT(f)+1$ $(DT_0(f), DT_1(f))$ denote the minimal number of leaves (0-leaves,1-leaves) in a decision tree for f. Let M(f) be the minimal number of monoms in a disjunctive form for f.

M(f) is also the complexity of f in Σ_2-circuits, that means circuits of depth 2 where the last level consists of an $\vee$-gate. The following result shows that decision trees are less efficient than depth-2 circuits.

Theorem 1: i) $DT^*(f) \geq DT_0(f) + DT_1(f)$.
ii) $DT_1(f) \geq M(f)$.
iii) $DT_0(f) \geq M(\bar{f})$.

Proof: i) is obvious.
ii) Consider a decision tree for f with the minimal number of 1-leaves. Any 1-leaf L corresponds to a unique path from the root to L. Let m(L) be the monom consisting of all variables and negated variables which must be 1 if we follow this path. Then f is the disjunction of all m(L) and we obtain a disjunctive form for f with $DT_1(f)$ monoms.
iii) follows in a similar way. Q.E.D.

The reader is asked to convince himself that the number of 1-leaves of