Texts in Theoretical Computer Science
An EATCS Series

Springer-Verlag Berlin Heidelberg GmbH

Peter Clote • Evangelos Kranakis

Boolean Functions and Computation Models

With 19 Figures

Springer

Authors

Prof. Dr. Peter Clote
Boston College
Department of Computer Science
and Department of Biology
Fulton Hall 410 B
140 Commonwealth Avenue
Chestnut Hill, MA 02467, USA
clote@cs.bc.edu

Prof. Dr. Evangelos Kranakis
Carleton University
School of Computer Science
1125 Colonel By Drive
Ottawa, Ontario, K1S 5B6, Canada
kranakis@scs.carleton.ca

Series Editors

Prof. Dr. Wilfried Brauer
Institut für Informatik
Technische Universität München
Arcisstrasse 21, 80333 München, Germany
Brauer@informatik.tu-muenchen.de

Prof. Dr. Grzegorz Rozenberg
Leiden Institute of Advanced Computer Science
University of Leiden
Niels-Bohrweg 1, 2333 CA Leiden, The Netherlands
rozenber@liacs.nl

Prof. Dr. Arto Salomaa
Turku Centre for Computer Science
Lemminkäisenkatu 14 A, 20 520 Turku, Finland
asalomaa@utu.fi

Library of Congress Cataloging-in-Publication Data

Clote, Peter.
Boolean functions and computation models/Peter Clote, Evangelos Kranakis.
p. cm. - (Texts in theoretical computer science)
Includes bibliographical references and index.

1. Computational complexity. 2. Algebra, Boolean. I. Kranakis, Evangelos. II. Title.
III. Series.

QA267.7 .C58 2001
511.3-dc21 2201031128

ACM Computing Classification (1998): F.1.1, F.4.1, F.1.3

ISBN 978-3-642-08217-7 ISBN 978-3-662-04943-3 (eBook)
DOI 10.1007/978-3-662-04943-3

Originally published by Springer-Verlag Berlin Heidelberg New York in 2002.
Softcover reprint of the hardcover 1st edition 2002

Cover Design: KünkelLopka, Heidelberg
Typesetting: Camera-ready by the authors
Printed on acid-free paper SPIN 10074467 45/3142SR - 5 4 3 2 1 0

Dedicated to our parents:

Mary Ann and Paul J. Clote
Stamatia and Kostantinos Kranakis

Preface

The foundations of computational complexity theory go back to Alan Turing in the 1930s who was concerned with the existence of automatic procedures deciding the validity of mathematical statements. The first example of such a problem was the undecidability of the Halting Problem which is essentially the question of debugging a computer program: Will a given program eventually halt? Computational complexity today addresses the quantitative aspects of the solutions obtained: Is the problem to be solved tractable? But how does one measure the intractability of computation? Several ideas were proposed: A. Cobham [Cob65] raised the question of what is the right model in order to measure a "computation step", M. Rabin [Rab60] proposed the introduction of axioms that a complexity measure should satisfy, and C. Shannon [Sha49] suggested the boolean circuit that computes a boolean function.

However, an important question remains: What is the nature of computation? In 1957, John von Neumann [vN58] wrote in his notes for the Silliman Lectures concerning the nature of computation and the human brain that

> ... logics and statistics should be primarily, although not exclusively, viewed as the basic tools of 'information theory'. Also, that body of experience which has grown up around the planning, evaluating, and coding of complicated logical and mathematical automata will be the focus of much of this information theory. The most typical, but not the only, such automata are, of course, the large electronic computing machines.
>
> Let me note, in passing, that it would be very satisfactory if one could talk about a 'theory' of such automata. Regrettably, what at this moment exists — and to what I must appeal — can as yet be described only as an imperfectly articulated and hardly formalized 'body of experience'.

With almost a half century after von Neumann's death, the theory of computation and automata is now a well-developed and sophisticated branch of mathematics and computer science. As he forecasted, the principal tools have proven to come from the fields of mathematical logic, combinatorics, and probability theory.

Using these tools, we have attempted to give a survey of the present state of research in the study of boolean functions, formulas, circuits, and propositional proof systems. All of these subjects are related to the overriding concern of how computation can be modeled, and what limitations and interrelations there are between different computation models.

This text is structured as follows. We begin with methods for the construction of boolean circuits which compute certain arithmetic and combinatorial functions, and investigate upper and lower bounds for circuit families. The techniques used are from combinatorics, probability and finite group theory. We then survey steps taken in a program initiated by S.A. Cook of investigating non-deterministic polynomial time, from a proof-theoretic viewpoint. Specifically, lower bounds are presented for lengths of proofs for families of propositional tautologies, when proven in certain proof systems. Techniques here involve both logic and finite combinatorics and are related to constant depth boolean circuits and to monotone arithmetic circuits.

Outline of the book

A more detailed breakdown of the book is as follows. In Chapter 1, circuits are constructed for data processing (string searching, parsing) and arithmetic (multiplication, division, fast Fourier transform). This material is intended to provide the reader with concrete examples, before initiating a more abstract study of circuit depth and size.

Chapter 2 presents a sampling of techniques to prove size lower bounds for certain restricted classes of circuits – constant depth or monotonic. These include Razborov's elegant constructive proof of the Håstad switching lemma, the Haken–Cook monotonic real circuit lower bound for the *broken moskito screen* problem, Razborov's algebraic approximation method for *majority*, and Smolensky's subsequent generalization to finite fields.

Chapter 3 studies symmetric boolean functions and related invariance groups. A characterization is given of those symmetric functions computable by constant depth polysize circuits. Invariance groups of boolean functions are characterized by a condition concerning orbit structure, and tight upper bounds are given for almost symmetric functions. Applications are given to anonymous networks such as rings and hypercubes. Most of these results are due to P. Clote and E. Kranakis.

Chapter 4 concerns the empirically observed *threshold* phenomenon concerning clause density $r = \frac{m}{n}$, where with high probability random formulas in k-CNF form having m clauses over n variables are satisfiable (unsatisfiable) if r is less (greater) than a threshold limit. The results of this chapter include a proof of an analytic upper bound, a result due to M. Kirousis, E. Kranakis and D. Krizanc.

Chapter 5 studies propositional proof systems, which have relevance to complexity theory, since $\mathrm{NP} = co{-}\mathrm{NP}$ if and only if there exists a polynomially bounded propositional proof system. In obtaining exponential lower bounds for increasingly stronger proof systems, new techniques have been developed, such as random restriction, algebraic and bottleneck counting methods – these techniques may ultimately play a role in separating complexity classes, and in any case are of interest in themselves. The proof systems include resolution, cutting planes, threshold logic, Nullstellensatz system, polynomial calculus, constant depth Frege, Frege, extended Frege, and substitution Frege systems.

In Chapter 6 we define various computation models including uniform circuit families, Turing machines and parallel random access machines, and illustrate some features of parallel computation by giving example programs. We then give characterizations of different parallel and sequential complexity classes in terms of function algebras – i.e., as the smallest class of functions containing certain initial functions and closed under certain operations. In the early 1960's, A. Cobham first defined polynomial time P and argued its robustness on the grounds of his machine independent characterization of P via function algebras.

With the development that certain programming languages now admit polymorphism and higher type functionals, using function algebras, complexity theory can now be lifted in a natural manner to higher types, a development which is the focus of Chapter 7. In that chapter, a new yet unpublished characterization of type 2 NC^1 functionals (due to the first author) is given in terms of a natural function algebra and related lambda calculus.

How to use the book

This text is to be of use to students as well as researchers interested in the emerging field of *logical complexity theory* (also called *implicit complexity theory*). The chapters of the book can be read as independent units. However one semester courses can be given as follows:

Semester Course	**Chapters**
Boolean Functions & Complexity	1, 2, 3
Proof Systems & Satisfiability	5, 4
Machine Models, Function Algebras & Higher Types	6, 7

At the end of every chapter, there are several exercises: some are simple extensions of results in the book while others constitute the core result of a research article. The various levels of difficulty are indicated with an asterisk placed before more difficult problems, and two asterisks for quite challenging and sometimes open research problems. The reader is invited to attempt them all.

Acknowledgments

Writing this book would have been impossible without the financial support of various research foundations, and without the exchange of ideas from many colleagues and friends.

Peter Clote is indebted to the NSF (National Science Foundation), CNRS (Centre National pour la Recherche Scientifique), Czech Academy of Science and Volkswagen Foundation for financial support of work on this text. In particular, thanks to J.-P. Ressayre for arranging a visit to Université Paris VII, and to D. Thérien for arranging a visit to the Barbados Complexity Theory Workshop, where some of the material from this text was presented. Evangelos Kranakis is indebted to NSERC (Natural Sciences and Engineering Research Council of Canada), and NWO (Netherlands Organization for the Advancement of Research) for partial support during the preparation of the book.

While holding the Gerhard Gentzen Chair of Theoretical Computer Science at the University of Munich, the first author (P. Clote) gave several courses using parts of the current text and would like to thank his students for the feedback. We would like to thank A. Abel, D. Achlioptas, T. Altenkirch, P. Beame, S. Bellantoni, E. Ben-Sasson, S. Buss, N. Danner, M. Hofmann, R. Impagliazzo, J. Johannsen, J. Krajíček, L. M. Kirousis, D. Krizanc, K.-H. Niggl, P. Pudlák, H. Schwichtenberg, Y. Stamatiou, T. Strahm, H. Straubing, G. Takeuti and J. Woelki for comments and suggestions, although of course the authors are solely responsible for any remaining errors. In particular, any omitted or erroneous references are purely unintentional. We are deeply grateful to Sam Buss, Jan Krajíček, Pavel Pudlák, and Gaisi Takeuti, with whom the first author collaborated over a period of years, and who have established many of the deepest results in propositional proof systems, as well as L. M. Kirousis and D. Krizanc with whom the second author has spent many enjoyable discussions.

Finally, we would like to express our deepest appreciation to Dr. Hans Wössner, Executive Editor for Computer Science of Springer-Verlag, who never lost faith in our project.

This book was type set using LaTeX with additional macros developed by S.R. Buss for typesetting proof figures.

Boston
Ottawa
July 2002

Peter Clote
Evangelos Kranakis

Contents