Grundlehren der mathematischen Wissenschaften 315

A Series of Comprehensive Studies in Mathematics

Springer-Verlag Berlin Heidelberg GmbH

Peter Bürgisser
Michael Clausen
M. Amin Shokrollahi

Algebraic Complexity Theory

With the Collaboration of Thomas Lickteig

With 21 Figures

Springer-Verlag Berlin Heidelberg GmbH

Peter Bürgisser
Institut für Mathematik
Abt. Angewandte Mathematik
Universität Zürich-Irchel
Winterthurerstraße 190
CH-8057 Zürich, Switzerland
buerg@amath.unizh.ch

Michael Clausen
Institut für Informatik V
Universität Bonn
Römerstraße 164
D-53117 Bonn, Germany
clausen@cs.uni-bonn.de

Mohammad Amin Shokrollahi
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704-1105, USA
and
Institut für Informatik V
Universität Bonn
Römerstraße 164
D-53117 Bonn, Germany
amin@icsi.berkeley.edu

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek – CIP-Einheitsaufnahme
Bürgisser, Peter: Algebraic complexity theory / Peter Bürgisser; Michael Clausen; M. Amin Shokrollahi. With the collab. of Thomas Lickteig. – Berlin; Heidelberg; New York; Barcelona; Budapest; Hong Kong; London; Milan; Paris; Santa Clara; Singapore; Tokyo: Springer 1997
(Grundlehren der mathematischen Wissenschaften; 315)

NE: Clausen, Michael:; Shokrollahi, Mohammad Amin:; GT

Mathematics Subject Classification (1991): 68Qxx, 05-xx, 14A10, 14P10, 15-xx, 16A46, 20Cxx, 60C05, 65Fxx, 65T10

ISBN 978-3-642-08228-3 ISBN 978-3-662-03338-8 (eBook)
DOI 10.1007/978-3-662-03338-8

Softcover reprint of the hardcover 1st edition 1997

Cover design: MetaDesign plus GmbH, Berlin
Typesetting: Authors' input files edited and reformatted by Kurt Mattes, Heidelberg, using a Springer TEX macro-package
SPIN: 10521707 41/3143-5 4 3 2 1 0 Printed on acid-free paper

To

Brigitte
Claudia, Julia, Simone
and Dorothe

Der Zweifler

Immer wenn uns
Die Antwort auf eine Frage gefunden schien
Löste einer von uns an der Wand die Schnur der alten
Aufgerollten chinesischen Leinwand, so daß sie herabfiel und
Sichtbar wurde der Mann auf der Bank, der
So sehr zweifelte.

Ich, sagte er uns
Bin der Zweifler, ich zweifle, ob
Die Arbeit gelungen ist, die eure Tage verschlungen hat.
Ob was ihr gesagt, auch schlechter gesagt, noch für einige Wert hätte.
Ob ihr es aber gut gesagt und euch nicht etwa
Auf die Wahrheit verlassen habt dessen, was ihr gesagt habt.
Ob es nicht vieldeutig ist, für jeden möglichen Irrtum
Tragt ihr die Schuld. Es kann auch eindeutig sein
Und den Widerspruch aus den Dingen entfernen; ist es zu eindeutig?
Dann ist es unbrauchbar, was ihr sagt. Euer Ding ist dann leblos.
Seid ihr wirklich im Fluß des Geschehens? Einverstanden mit
Allem, was wird? Werdet ihr noch? Wer seid ihr? Zu wem
Sprecht ihr? Wem nützt es, was ihr da sagt? Und nebenbei:
Läßt es auch nüchtern? Ist es am Morgen zu lesen?
Ist es auch angeknüpft an Vorhandenes? Sind die Sätze, die
Vor euch gesagt sind, benutzt, wenigstens widerlegt? Ist alles belegbar?
Durch Erfahrung? Durch welche? Aber vor allem
Immer wieder vor allem andern: Wie handelt man
Wenn man euch glaubt, was ihr sagt? Vor allem: Wie handelt man?

Nachdenklich betrachteten wir mit Neugier den zweifelnden
Blauen Mann auf der Leinwand, sahen uns an und
Begannen von vorne.

Bertolt Brecht

Preface

The algorithmic solution of problems has always been one of the major concerns of mathematics. For a long time such solutions were based on an intuitive notion of algorithm. It is only in this century that metamathematical problems have led to the intensive search for a precise and sufficiently general formalization of the notions of computability and algorithm.

In the 1930s, a number of quite different concepts for this purpose were proposed, such as Turing machines, WHILE-programs, recursive functions, Markov algorithms, and Thue systems. All these concepts turned out to be equivalent, a fact summarized in Church's thesis, which says that the resulting definitions form an adequate formalization of the intuitive notion of computability. This had and continues to have an enormous effect. First of all, with these notions it has been possible to prove that various problems are algorithmically unsolvable. Among these undecidable problems are the halting problem, the word problem of group theory, the Post correspondence problem, and Hilbert's tenth problem. Secondly, concepts like Turing machines and WHILE-programs had a strong influence on the development of the first computers and programming languages.

In the era of digital computers, the question of finding efficient solutions to algorithmically solvable problems has become increasingly important. In addition, the fact that some problems can be solved very efficiently, while others seem to defy all attempts to find an efficient solution, has called for a deeper understanding of the intrinsic computational difficulty of problems. This has resulted in the development of complexity theory. Complexity theory has since become a very diversified area of research. Each branch uses specific models of computation, like Turing machines, random access machines, Boolean circuits, straight-line programs, computation trees, or VLSI-models. Every computation in such a model induces costs, such as the number of computation steps, the amount of memory required, the number of gates of a circuit, the number of instructions, or the chip area. Accordingly, studies in computational complexity are generally based on some model of computation together with a complexity measure. For an overview, we refer the interested reader to the *Handbook of Theoretical Computer Science* [321], which contains several surveys of various branches of complexity theory.

In this book we focus on *Algebraic Complexity Theory*, the study of the intrinsic algorithmic difficulty of algebraic problems within an algebraic model of computation. Motivated by questions of numerical and symbolic computation, this branch of research originated in 1954 when Ostrowski [403] inquired about the optimality of Horner's rule. Algebraic complexity theory grew rapidly and has since become a well-established area of research. (See the surveys of von zur Gathen [189], Grigoriev [210], Heintz [241], Schönhage [462], and Strassen [506, 510].) However, with the exception of the now classic monograph by Borodin and Munro [65], published in 1975, a systematic treatment of this theory is not available.

This book is intended to be a comprehensive text which presents both traditional material and recent research in algebraic complexity theory in a coherent way. Requiring only some basic algebra and offering over 350 exercises, it should be well-suited as a textbook for beginners at the graduate level. With its extensive bibliographic notes covering nearly 600 research papers, it might also serve as a reference book.

The text provides a uniform treatment of algebraic complexity theory on the basis of the straight-line program and the computation tree models, with special emphasis on *lower complexity bounds*. This also means that this is not a book on Computer Algebra, whose main theme is the design and implementation of efficient algorithms for algebraic problems.

Nonetheless, our book contains numerous algorithms, typically those that are essentially optimal within the specified computation model. Our main goal is to develop methods for proving the optimality of such algorithms.

To emphasize the logical development of the subject, we have divided the book into five parts, with 21 chapters in total. The first chapter consists of an informal introduction to algebraic complexity theory.

The next two chapters form PART I: FUNDAMENTAL ALGORITHMS. Chapter 2 is concerned with efficient algorithms for the symbolic manipulation of polynomials and power series, such as the Schönhage-Strassen algorithm for polynomial multiplication, the Sieveking-Kung algorithm for the inversion of power series, or the Brent-Kung algorithm for the composition of power series. It is followed by a chapter in which the emphasis lies on efficient algorithms within the branching model. In particular, we present the fast Knuth-Schönhage algorithm for computing the greatest common divisor (GCD) of univariate polynomials. This algorithm combined with Huffman coding then yields efficient solutions of algorithmic problems associated with Chinese remaindering. Furthermore the VC-dimension and the theory of epsilon nets are used to show that certain **NP**-complete problems, like the knapsack or the traveling salesman problem, may be solved by "nonuniform polynomial time algorithms" in the computation tree model over the reals. This surprising and important result, due to Meyer auf der Heide, demonstrates that it is not possible to prove exponential lower bounds for the above problems in the model of computation trees. Moreover, it stresses the role of uniformity in the definition of the language class **NP** and, at the same time, puts emphasis on the quality of several lower bounds derived later in Chapter 11.

While the first three chapters rely on the reader's intuitive notion of algorithm, the remaining parts of the book, directed towards lower bounds, call for an exact specification of computation models and complexity measures.

Therefore, in PART II: ELEMENTARY LOWER BOUNDS (Chapters 4–7), we first introduce the models of straight-line programs and computation trees, which we use throughout the rest of the book. We then describe several elementary lower bound techniques. Chapter 5 contains transcendence degree arguments, including results of Motzkin and Belaga as well as the Baur-Rabin theorem. Chapter 6 discusses a unified approach to Pan's substitution method and its extensions. The methods of Chapters 5 and 6 yield lower bounds which are at most linear in the number of input variables. Nonetheless, the methods are strong enough to show the optimality of some basic algorithms, the most prominent being Horner's rule. In Chapter 7 we introduce two fundamental program transformation techniques. The first is Strassen's technique of "avoiding divisions." The second is a method for transforming a program for the computation of a multivariate rational function into one which computes the given function *and* all its first-order partial derivatives. The results of Chapter 7 are of importance in Chapters 8, 14, and 16.

PART III: HIGH DEGREE (Chapters 8–12) shows that concepts from algebraic geometry and algebraic topology, like the degree or Betti numbers, can be applied to prove nonlinear lower complexity bounds. Chapter 8 studies Strassen's degree bound, one of the central tools for obtaining almost sharp lower complexity bounds for a number of problems of high degree, like the computation of the coefficients of a univariate polynomial from its roots. Chapter 9 is devoted to the investigation of specific polynomials that are hard to compute. It may be considered as a counterpart to Chapters 5 and 6 where we study generic polynomials. In Chapter 10 the degree bound is adapted to the computation tree model. With this tool it turns out that the Knuth-Schönhage algorithm is essentially optimal for computing the Euclidean representation. In Chapter 11 Ben-Or's lower complexity bound for semi-algebraic membership problems is deduced from the Milnor-Thom bound. This is applied to several problems of computational geometry. In Chapter 12 the Grigoriev-Risler lower bound for the additive complexity of univariate real polynomials is derived from Khovanskii's theorem on the number of real roots of sparse systems of polynomial equations.

PART IV: LOW DEGREE (Chapters 13–20) is concerned with the problem of computing a finite set of multivariate polynomials of degree at most two. In Chapter 13 we discuss upper and lower complexity bounds for computing a finite set of linear polynomials, which is simply the task of multiplying a generic input vector by a specific matrix. This problem is of great practical interest, as the notable examples of the discrete Fourier transform (DFT), Toeplitz, Hankel and Vandermonde matrices indicate.

The theory of bilinear complexity is concerned with the problem of computing a finite set of bilinear polynomials. Chapters 14–20 contain a thorough treatment of this theory and can be regarded as a book within a book. Chapter 14 introduces the framework of bilinear complexity theory and is meant as a prerequisite for Chapters 15–20. The language introduced in Chapter 14 allows a concise discussion of the matrix multiplication methods in Chapter 15, such as Strassen's original algorithm and the notion of rank, Bini-Capovani-Lotti-Romani's concept of border rank, Schönhage's τ-theorem, as well as Strassen's laser method, and its tricky extension by Coppersmith and Winograd. Chapter 16 shows that several problems in computational linear algebra are about as hard as matrix multiplication, thereby emphasizing the key role of the matrix multiplication problem. Chapter 17 discusses Lafon and Winograd's lower bound for the complexity of matrix multiplication, and its generalization by Alder and Strassen. Moreover, in Chapter 18 we study a relationship, observed by Brockett and Dobkin, between the complexity of bilinear maps over finite fields and a well-known problem of coding theory. Partial solutions to the latter lead to interesting lower bounds, some of which are not known to be valid over infinite fields. This chapter also discusses the Chudnovsky-Chudnovsky interpolation algorithm on algebraic curves which yields a linear upper complexity bound for the multiplication in finite fields.

The bilinear complexity or rank of bilinear problems can be reformulated in terms of tensors, resulting in a generalization of the usual matrix rank. In Chapter 19 tensorial rank is investigated for special classes of tensors, while Chapter 20 is devoted to the study of the rank of "generic" tensors. In the language of algebraic geometry this problem is closely related to computing the dimension of higher secant varieties to Segre varieties.

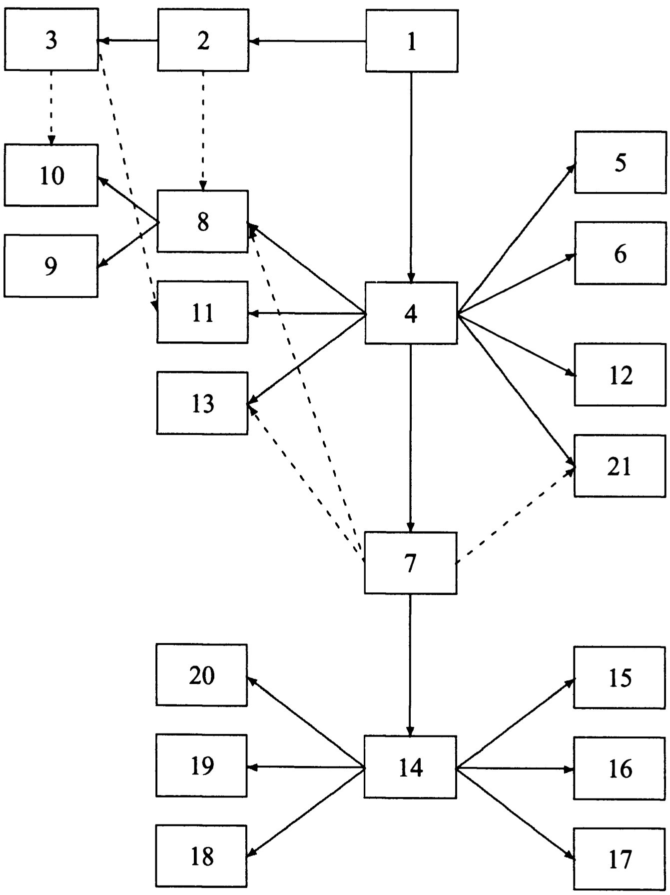
PART V: COMPLETE PROBLEMS (Chapter 21) presents Valiant's nonuniform algebraic analogue of the **P** versus **NP** problem. It builds a bridge both to the theory of **NP**- and #**P**-completeness as well as to that part of algebraic complexity theory which is based on the parallel computation model.

A number of topics are not covered in this book; this is due to limitations of time and space, the lack of reasonable lower complexity bounds, as well as the fact that certain problems do not fit into the straight-line program or computation tree model. More specifically, our book treats neither computational number theory nor computational group and representation theory (cf. Cohen [117], Lenstra and Lenstra [326], Sims [484], Atkinson (ed.) [13], Lux and Pahlings [344], Finkelstein and Kantor (eds.) [172]). Also, we have not included a discussion of topics in computational commutative algebra like factorization and Gröbner bases, nor do we speak about the complexity of first-order algebraic theories (cf. Becker and Weispfenning [34], Fitchas et al. [174], Heintz et al. [245], and Kaltofen [284, 286]). We have also omitted a treatment of parallel and randomized algorithms (cf. von zur Gathen [186], Ja'Ja [268]). However, many of these topics have already been discussed in other books or surveys, as the given references indicate.

Clearly, much is left to be done. We hope that our book will serve as a foundation for advanced research and as a starting point for further monographs on algebraic complexity theory.

Zürich, Bonn, and Berkeley *P. Bürgisser · M. Clausen · M.A. Shokrollahi*
June 1996

Leitfaden



Notes to the Reader

This book is intended as a textbook as well as a reference book. One of the important principal features is the division of the material into the relatively large number of 21 chapters, which are each designed to enable quick acquaintance with a specific topic. Furthermore, we have subdivided each chapter into sections which often make widely differing demands on the reader. Almost every chapter starts at an undergraduate level and ends at a more advanced level. To facilitate the reader's orientation we have marked those sections with asterisks that are of a rather technical nature and may be skipped on a first reading. To provide easy checks on the reader's comprehension of the text, or to challenge her/his proficiency, we have included numerous exercises in each chapter, the harder ones carrying asterisks. Many of the exercises are important results in their own right and are occasionally referred to in later sections. A list of open problems as well as the detailed notes at the end of each chapter should be seen not only as incentives for researchers willing to improve the present knowledge, but also as landmarks pointing to the frontiers of our field.

We believe that the structure of the book facilitates its use in many ways. Generally, all readers interested in lower complexity bounds are expected to study the essential material of Sections 4.1–4.2, where we describe straight-line programs and introduce the notion of complexity. The language developed there will be used throughout the book. Thereafter, those whose primary inclination is to use this book as a reference source can directly traverse to their topic of interest.

The rigorous presentation of many techniques for lower bound proofs in algebraic complexity theory calls not only for the use of tools from different areas of mathematics, but also for technicalities which often obscure the ideas behind those techniques. Whenever we have encountered such a situation, we have tried to familiarize the reader with the underlying ideas by means of examples of increasing difficulty. In so doing, we have designed a textbook for various possible courses. As an example of an introductory course on algebraic complexity theory, one can cover the topics presented in (1) (where (x) means "parts of Chapter x"), 2, 4.1–4.2, 5, 6, 7.2, 8.1. This course could be followed by an advanced course dealing with the content of (1), 4.4–4.5, 3.1–3.2, 8.2–8.5, 10.1–10.2, 11. A special course on bilinear complexity could include (1), 4.1–4.2, 14, 15.1–15.8, 17.1–17.3, 19.1–19.2. A special course on the Degree Bound might consist of (1), (2), (4), 7.2, 8.2–8.4, 3.1–3.2, 10.1–10.2, (11).

Isolated chapters of our book can be used by people from other disciplines as complementary material to courses in their own field of research. Examples of this include courses on **NP**-completeness + (21), coding theory + (18), group representation theory + (13), computational geometry + (11), algebraic number theory + 9.1–9.3, and numerical analysis + (5, 6, 7, 8, 16). Courses in computer algebra can obviously be accompanied by a treatment of several of the lower complexity bounds discussed in this book. In addition, there is also a number of (asymptotically) fast algorithms in Chapters 2, 3, 5, 13, and 15 that are of interest to computer algebraists.

Acknowledgments

Our greatest intellectual debt is to V. Strassen for his many contributions to the field of algebraic complexity theory as well as for his brilliant lectures which introduced the subject to us. Special thanks go to our cooperator Thomas Lickteig who, together with us, first planned this book more than five years ago. His competence in this field has always been of extreme benefit to us. We owe thanks to W. Baur whose clear and concise lecture notes helped us a lot in writing this book.

We are indebted to Ch. Bautz, F. Bigdon, A. Björner, K. Kalorkoti, F. Mauch, M. Nuesken, T. Recio, H. J. Stoß, V. Strassen and Ch. Zengerling for reading parts of the manuscript and their valuable suggestions for improvements. We have benefited from the help of U. Baum, S. Blackburn, J. Buhler, E. Kaltofen, H. Meier-Reinhold, A. McNeil, J. Neubüser, A. Schönhage, and F. Ulmer and would like to express our gratitude to them. We also thank our students at the Universities of Bonn and Zürich for their attention and stimulating questions.

Although this book has been proofread by several people, we take complete responsibility for the errors that may have remained.

Many people, too numerous to mention, have contributed to our project by kindly sending to us a list of their publications relevant for our book. We thank them all very much.

We thank the Schweizerische Nationalfonds for its financial support which allowed the first author to stay at the University of Bonn in the first phase of our project from 1991 until 1993. Thanks go also to the Institute of Applied Mathematics of the University of Zürich for the pleasant working conditions which allowed an efficient continuation of the project after the first author had moved to Zürich.

We have extensively used Email and Internet, mostly after the first and third author had left Bonn for Zürich and Berkeley, respectively. Without these media, communication would have become much harder. Also, we have benefited a lot from the GNU project, in particular from the powerful Emacs-Editor distributed with the GNU-package. We take the opportunity to thank R. Stallman and his team for this public domain software of distinguished quality.

Without the document processing systems TeX and LaTeX we would have had a very hard time. Many thanks to D. Knuth and L. Lamport for providing the community with their wonderful – and free – software. For the camera-ready preparation of this document we have used different style files written by B. Althen, M. Barr, and P. Taylor, whom we would like to thank.

We are especially grateful to the staff at Springer-Verlag Heidelberg for their editorial advise and great patience throughout this enterprise.

Finally, we wish to thank Brigitte, Claudia, and Dorothe for their support, patience, and understanding of the commitment necessary to write such a book.

Table of Contents

Part IV. Low Degree