Monographs in Computer Science

Editors

David Gries
Fred B. Schneider

Monographs in Computer Science

Abadi and Cardelli, **A Theory of Objects**

Benosman and Kang [editors], **Panoramic Vision: Sensors, Theory, and Applications**

Broy and Stølen, **Specification and Development of Interactive Systems: FOCUS on Streams, Interfaces, and Refinement**

Brzozowski and Seger, **Asynchronous Circuits**

Burgin, **Super-Recursive Algorithms**

Cantone, Omodeo, and Policriti, **Set Theory for Computing: From Decision Procedures to Declarative Programming with Sets**

Castillo, Gutiérrez, and Hadi, **Expert Systems and Probabilistic Network Models**

Downey and Fellows, **Parameterized Complexity**

Feijen and van Gasteren, **On a Method of Multiprogramming**

Herbert and Spärck Jones [editors], **Computer Systems: Theory, Technology, and Applications**

Leiss, **Language Equations**

Levin, Heydon, and Mann, **Software Configuration Management with VESTA**

McIver and Morgan [editors], **Programming Methodology**

McIver and Morgan [editors), **Abstraction, Refinement and Proof for Probabilistic Systems**

Misra, **A Discipline of Multiprogramming: Programming Theory for Distributed Applications**

Nielson [editor], **ML with Concurrency**

Paton [editor], **Active Rules in Database Systems**

Selig, **Geometrical Methods in Robotics**

Selig, **Geometric Fundamentals of Robotics, Second Edition**

Shasha and Zhu, **High Performance Discovery in Time Series: Techniques and Case Studies**

Tonella and Potrich, **Reverse Engineering of Object Oriented Code**

Mark Burgin

Super-Recursive Algorithms

Springer

Mark Burgin
Department of Mathematics
UCLA
Los Angeles, CA 90095
U.S.A.
mburgin@math.ucla.edu

Series Editors:
David Gries
Cornell University
Department of Computer Science
Ithaca, NY 14853
U.S.A.

Fred B. Schneider
Cornell University
Department of Computer Science
Ithaca, NY 14853
U.S.A.

Library of Congress Cataloging-in-Publication Data
Burgin, M.S. (Mark Semenovich)
Super-recursive algorithms / Mark Burgin.
p. cm. — (Monographs in computer science)
Includes bibliographical references and index.
ISBN 0-387-95569-0 (alk. paper)
1. Recursive functions. 2. Algorithms. I. Title. II. Series.

QA9.615.B87 2005
511.3′52—dc22 2004041748

ISBN 0-387-95569-0 Printed on acid-free paper.

Printed in the United States of America. (TXQ/MV)

9 8 7 6 5 4 3 2 1 SPIN 10891097

springeronline.com

Super-Recursive Algorithms

To my parents and grandparents

Contents

Preface ix

1 Introduction 1
1.1 Information processing systems (IPS) 2
1.2 What theory tells us about new directions in information technology 8
1.3 The structure of this book 12
1.4 Notation and basic definitions 17

2 Recursive Algorithms 21
2.1 What algorithms are and why we need them 21
2.2 Mathematical models of algorithms and why we need them: History and methodology 32
2.3 Centralized computation: Turing machines 46
2.4 Distributed computation: Neural networks and cellular automata 56
2.5 Applications 72

3 Subrecursive Algorithms 79
3.1 What subrecursive algorithms are and why we need them 79
3.2 Mathematical models of subrecursive algorithms and why we need them 81
3.3 Procedural programming as know-how: Finite automata and finite-state machines 85
3.4 Functional programming as know-what: Recursive functions 97

4 Superrecursive Algorithms: Problems of Computability 107
4.1 What superrecursive algorithms are and why we need them 108
4.2 Mathematical models of superrecursive algorithms and why we need them 114
4.3 Emerging computation, inductive Turing machines, and their computational power 152
4.4 Grid automata: Interaction and computation 184

5 Superrecursive Algorithms: Problems of Efficiency 203
5.1 Measures of computer efficiency, program complexity, and their mathematical models 204
5.2 Computational complexity: Axiomatic approach and specific measures 212
5.3 Dual complexity measures: Axiomatic approach and Kolmogorov complexity 227

6 Conclusion: Problems of Information Technology and Computer Science Development 245
6.1 A systemology for models of algorithms and information processing systems 247
6.2 Development of information processing systems (IPS) 251
6.3 From algorithms to programs to technology 258

References and Sources for Additional Reading 263

Index 295

Preface

Progress is impossible without change,
and those who cannot change their minds
cannot change anything.
George Bernard Shaw (1856–1950)

Any sufficiently advanced technology is
indistinguishable from magic.
Arthur C. Clarke (1917–)

This book introduces the new realm of superrecursive algorithms and the development of mathematical models for them. Although many still believe that only recursive algorithms exist and that only some of them are realizable, there are many situations in which people actually work with superrecursive algorithms. Examples of models for superrecursive algorithms are abstract automata like inductive Turing machines as well as computational schemes like limiting recursive functions.

The newly emerging field of the theory of superrecursive algorithms belongs to both mathematics and computer science. It gives a glimpse into the future of computers, networks (such as the Internet), and other devices for information interchange, processing, and production. In addition, superrecursive algorithms provide more adequate models for modern computers, the Internet, and embedded systems. Consequently, we hope (and expect) that this theory of superrecursive algorithms will, in the end, provide new insight and different perspectives on the utilization of computers, software, and the Internet.

The first goal of this book is to explain how superrecursive algorithms open new kinds of possibilities for information technology. This is an urgent task. As Papadopoulos (2002) writes, "If we don't rethink the way we design computers, if we don't find new ways of reasoning about distributed systems, we may find ourselves eating sand when the next wave hits." We believe that a theory of superrecursive algorithms makes it possible to introduce a new paradigm for computation, one that yields better insight into future functioning of computers and networks. This form of computation will eclipse the more familiar kinds and will be commercially available before exotic technologies such as DNA and quantum computing arrive.

Another goal of this book is to explain how mathematics has explicated and evaluated computational possibilities and its role in extending the boundaries of computation. As we do this, we will present the theory of algorithms and computation in a new, more organized structure.

It is necessary to remark that there is an ongoing synthesis of computation and communication into a unified process of information processing. Practical and theoretical advances are aimed at this synthesis and also use it as a tool for further development. Thus, we use the word computation in the sense of information processing as a whole. Better theoretical understanding of computers, networks, and other information processing systems will allow us to develop such systems to a higher level. As Terry Winograd (1997) writes, "The biggest advances will come not from doing more and bigger and faster of what we are already doing, but from finding new metaphors, new starting points." In this book, we attempt to show that such new metaphors already exist and that we need only to learn how to use them to extend the world of computers in ways previously unimaginable.

Algorithms and their theory are the basis of information technology. Algorithms have been used by people since the beginning of time. Algorithms rule computers. Algorithms are so important for computers that even the mistakes of computers result mostly from mistakes of algorithms in the form of software. Consequently, the term "algorithm" has become a general scientific and technological concept used in a variety of areas. The huge diversity of algorithms and their mathematical models builds a specific "algorithmic universe". However, the science that studies algorithms emerged only in the twentieth century.

Since the emergence of the theory of algorithms, mathematicians and computer scientists learned a lot. They have built mathematical models for algorithms and, by means of these models, discovered a principal law of the algorithmic universe, the Church–Turing thesis, and it governs the algorithmic universe just as Newton's laws govern our physical universe. However, as we know, Newton's laws are not universal. They are true for processes that involve only ordinary bodies. Einstein, Bohr, Dirac, and other great physicists of the twentieth century discovered more fundamental laws in the microworld that go beyond the scope of Newton's laws. In a similar way, new laws for the algorithmic universe have been discovered that go beyond the Church–Turing thesis. The Church–Turing thesis encompasses only a small part of the algorithmic universe, including recursive algorithms. This book demonstrates that superrecursive algorithms are more powerful, efficient, and tractable than recursive algorithms, and it introduces the reader to this new, expanded algorithmic universe.

Consider the famous Gödel theorem on the incompleteness of formal arithmetic. In the context of recursive algorithms, this theorem has absolute and ultimate meaning, vitally restricting the abilities of mathematicians and mathematics. In the context of superrecursive algorithms, the Gödel theorem becomes relative, stating only differences in abilities based on superrecursive and recursive algorithms. That is, the theory articulates that, for sufficiently rich mathematical theories, such as arithmetic, superrecursive algorithms allow one to prove much more than conventional methods of formal deduction, which are based on recursive algorithms (Burgin, 1987). When Gödel proved his theorem, it was a surprise to most mathematicians. However, from the superrecursive perspective, the Gödel theorem is a natural result that simply reflects the higher computational and deductive power of superrecursive algorithms.

Although the main concern of this book is superrecursive algorithms and hypercomputation, a variety of other problems are also analyzed. They include general problems such as: What is an algorithm? What is a description of an algorithm?

How do we measure computational power, computational efficiency, computational equivalency for computers, networks, and embedded systems and their mathematical models? What are the structures, types, and functioning of information processing systems? Can we provide a systematization of mathematical models of computational processes and their algorithmic representations? How do they affect computer, network, and information-processing architectures?

The organization of this book

This book begins with models of conventional, recursive algorithms, with an overview of the theory of recursive algorithms given in Chapter 2. We then present even less powerful, but more tractable and feasible, subrecursive algorithms, giving an overview of their theory in Chapter 3. We consider some classes of subrecursive algorithms that are determined by restrictions in construction; for instance, finite automata. Subrecursive algorithms defined by restrictions on the resources used, e.g., Turing machines with polynomial time of computation, are mentioned only tangentially.

Our approach has a three-fold aim. The first aim is to prepare a base for superrecursive algorithms; an exposition of conventional algorithms helps to understand better the properties and advantages of new and more powerful algorithmic patterns, superrecursive algorithms.

The second aim of our approach is to give a general perspective on the theory of algorithms. Computer scientists and mathematicians have elaborated a huge diversity of models. Here we try to systematize these models from a practical perspective of computers and other information processing systems. As far as we know, this is the first attempt of its kind.

The third aim of our approach is to achieve completeness, making the book self-contained. This allows a reader to understand the theory of algorithms and computation as a whole without going to other sources. Of course, other sources may be used for further studies of separate topics. For instance, Rogers (1967) has more material about recursive algorithms, and Hopcroft, Motwani, and Ullman (2001) contains more material about finite automata and context-free grammars.

But this book allows a reader to better comprehend other theories in computer science by systematizing them, extending their scope, and developing a more advanced perspective based on superrecursive algorithms.

After considering conventional models of algorithms, we introduce models of superrecursive algorithms. In Chapter 4 we consider the computational power of superrecursive algorithms. In Chapter 5 we consider the efficiency of superrecursive algorithms, which is represented in the theory by a kind of complexity measure. The book culminates in a positive reevaluation of the future development of communication and information processing systems.

The exposition is aimed at different groups of readers. Those who want to know more about the history of computer science and get a general perspective of the current situation in the theory of algorithms and its relations to information technology can skip proofs and even many results that are given in the strict mathematical form. At the same time, those who have a sufficient mathematical training and are interested mostly in computer and information science or mathematics can skip preliminary deliberations and go directly to the exposition of superrecursive algorithms and automata. Thus, a variety of readers will be able to find interesting and useful issues in this book.

It is necessary to remark that the research in this area is so active that it is impossible to include all ideas, issues, and references, for which we ask the reader's forbearance.

Theories that study information technology belong to three disciplines: information sciences, computer science, and communication science. All such theories have a mathematical foundation, so it is no surprise that mathematics has its theories of computers and computations. The main theory is the theory of algorithms, abstract automata, and computation, or simply, the theory of algorithms. It explains in a logical way how computers function and how computations are organized. It provides means for evaluation and development of computers, nets, and other computational systems and processes. For example, a search for new kinds of computing resulted in molecular (in particular, DNA) and quantum computing, which are the most widely discussed. At this point, however, both of these paradigms appear to be restricted to specialized domains (molecular for large combinatorial searches, quantum for cryptography) and there are no working prototypes of either. The theory of algorithms finds a correct place for them in a wide range of different computational schemes.

Acknowledgments

Many wonderful people have made contributions to my efforts with this work. I am especially grateful to Springer-Verlag and its editors for their encouragement and help in bringing about this publication. In developing ideas in the theory of algorithms, automata, and computation, I have benefited from conversations with many friends and colleagues who have communicated with me on problems of superrecursive algorithms, and I am grateful for their interest and help.

Credit for my desire to write this book must go to my academic colleagues. Their questions and queries made significant contributions to my understanding of algorithms and computation. I would particularly like to thank many fine participants of the Logic Colloquium of the Department of Mathematics, the Computer Science Department Seminar, the Applied Mathematics Colloquium, Seminar of Theoretical Computer Science of the Department of Computer Science at UCLA, and the CISM Seminar at JPL for extensive and helpful discussions of the theory of superrecursive algorithms, which gave me so much encouragement. Discussions with A.N. Kolmogorov from Moscow State University and Walter Karplus from UCLA gave much to the development of ideas and concepts of the theory of superrecursive algorithms. I would also like to thank the Departments of Mathematics and Computer Science in the School of Engineering at UCLA for providing space, equipment, and helpful discussions. Finally, teaching the theory of computation in the Department of Computer Science at UCLA has given me a wonderful opportunity to test many new ideas from the theory of algorithms.