THEORY OF COMPUTATION

Derick Wood

H7

THEORY OF COMPUTATION

Derick Wood

University of Waterloo

1817

HARPER & ROW, PUBLISHERS, New York
Cambridge, Philadelphia, San Francisco, Washington,
London, Mexico City, São Paulo, Singapore, Sydney

To Ma Li

Sponsoring Editor: John Willig
Project Editor: Thomas R Farrell
Cover Design: 20/20 Services, Inc.
Text Art: RDL Artset, Ltd.
Production: Willie Lane
Printer and Binder: R. R. Donnelley & Sons Company

Theory of Computation

Library of Congress Cataloging-in-Publication Data

Wood, Derick, 1940–
Theory of computation.

Bibliography: p.
Includes index.
1. Computational complexity. 2. Machine theory.
3. Electronic data processing—Mathematics. I. Title.
QA267.W66 1987 511 86-11976
ISBN 0-06-047208-1

86 87 88 89 9 8 7 6 5 4 3 2 1

Preface

This book is intended to be used as the basis of a one- or two-term introductory course in the theory of computation at the junior and senior levels. It concentrates on the fundamental models for languages and computation together with their properties. The models are finite automata, regular expressions, context-free grammars, pushdown automata, Turing machines, and transducers. The properties that are presented are family relationships (the relative power of the models), closure properties, decidability properties, and complexity properties. The book contains simple and elegant proofs of many results that are usually considered difficult. For example the proof that we give to show that every finite automaton has an equivalent regular expression is little known.

Throughout the text, algorithms are given in a PASCAL-like notation, since there is an emphasis on constructions and programming. In particular each of Chapters 0-7 has programming projects in addition to the more usual exercises. There is also an emphasis on practical applications throughout the text; for example, finite automata and pattern matching, regular expressions and text editing, extended context-free grammars and syntax diagrams, finite transducers and data compression. The text contains an abundance of worked examples to help the student understand the concepts as they are introduced.

Each chapter terminates with a summary of the material, its history, and a springboard; the last two items include references to the current literature. The springboard introduces a few topics for further investigation that can be used as projects for high-caliber students.

The text consists of four parts. Preliminaries are dealt with in Part I. Chapter 0 reviews sets, relations, graphs, functions, enumerability, and proof techniques. Chapter 1 introduces decision problems, formal languages, and their relationship.

Part II presents five models for languages and computation in Chapters 2-6: finite automata, regular expressions, context-free grammars, pushdown automata, and Turing machines. Chapters 2-6 are written to be order-independent to a large extent. In Chapter 7 these models are adapted for relations and functions. It presents finite transducers (from finite automata), translation grammars (from context-free grammars), pushdown transducers (from pushdown automata), and Turing transducers (from Turing machines). Turing transducers give rise to Turing-computable functions, the largest known class of computable functions. Finally, primitive recursive functions and recursive functions are introduced as an alternative method of specifying functions.

Part III covers the fundamental properties of the models. Chapter 8 explores the relative expressive power of the models, that is, family relationships. Chapter 9 investigates how instances of each model can be formed using a building block approach, that is, the closure properties of the associated language families. Decision problems for the models are studied in Chapter 10. Both basic decidable and undecidable problems are considered.

Part IV explores some further topics in a single chapter, Chapter 11. The topics are NP-completeness, general grammars, and parsing.

Lemmas and theorems are numbered sequentially within sections, in the same sequence. Examples are numbered sequentially within sections and continuations of the same example are indicated by using uppercase letters in addition. Exercises and programming projects are also numbered by section; programming projects are distinguished by an initial letter "P". Finally, an asterisk "*" and a double asterisk "**" indicate that a section or exercise is "difficult" and "more difficult", respectively; hence, the corresponding sections can be omitted on a first reading.

Many people have helped me to develop the present version of the text by giving their encouragement, their comments, or their methods of presenting material. These include Ray Bagley who reviewed the manuscript and made a large number of excellent suggestions; John Brzozowski, my colleague, who not only read and commented on an earlier version, but also by his teaching methods suggested many improvements; Patrick Dymond, who discussed the content of Chapters 6, 7, and 8; Nathan Friedman, a reviewer, who gave some excellent suggestions; Ed Gurari, another reviewer, who made many useful comments; Heikki Mannila, who was willing to participate in a discussion about a suggested approach at the drop of a hat; Areski Nait-Abdallah, who encouraged the germination of this text; Thomas Ottmann, who helped me in the development of Chapters 1 and 7; Jan van Leeuwen, who gave me some much needed positive response in the early days of the text; and Sheng Yu, who was willing to listen and comment on my mutterings. The many anonymous reviewers who waded through preliminary drafts have my sympathy as well as my thanks.

However, without the foundation of friendship and collaboration that I have enjoyed over many years with Hermann Maurer and Arto Salomaa I would not have been able to contemplate this book, let alone write it.

I am very grateful to Mary Chen, who typeset this book, for her concern, patience, and excellent work. She designed the layout, suggested type styles, and much more. I wish to thank Ian Allen, who was responsible for the macros that implemented her suggested layout. The excitement, continuous encouragement, and warmth of my editor, John Willig, my project editor, Thomas Farrell, and other staff members of Harper & Row, even when I fell behind schedule, was much appreciated.

Last, but not least, I tender thanks to the Lord of the Universe who enabled me to meet criticism and deadlines, to persevere, and to aim for excellence. Dei Gloria.

I am sure that errors remain to be found; however, they are mine and mine alone. I will be glad to hear of any errors that you find and any corrections or suggestions that you have.

Derick Wood

Course Outlines

This text contains much more than one term's worth of material; therefore, I give some suggested course outlines below. These reflect different emphases that an instructor may wish to give or that a syllabus requires. First, however, I wish to give some general guidelines to using this book.

The text has an unusual arrangement of material, in that a variety of models are presented in Part II and their properties are discussed mostly in Part III. This provides the basis of a course in which only models are emphasized; perhaps in a first course. It also allows the student to see similar results, concerning properties of models, side by side, making it easier for the student to compare and contrast proof techniques. One effect of this separation is that complete coverage of a standard topic such as finite automata and regular expressions will need not only material from Chapters 2 and 3, but also snippets from Chapters 8, 9, and 10 and also, possibly, from Chapter 7.

The text also contains an extensive review of discrete mathematics in Chapter 0. Experience has shown that students have little prior exposure to or understanding of graphs, relations, closure, and proof techniques. This will vary tremendously from university to university, therefore this is omitted from the course outlines given below. It is anticipated that students begin with Chapter 1, which introduces the basic questions of the theory of computaion, the basics of language theory, and also sketches an important connection between languages a[illegible]cision problems.

In a thirteen-week course at the Univ[illegible]ity of Waterlo[illegible]e have been able to cover finite automata, regular expressions, context-free grammars, Turing machines, closure properties, family relationships, and decidability properties. More recently we covered only finite automata, regular expressions, Turing machines, and their properties together with an introduction to context-free grammars. A sample thirteen-week schedule might be as follows:

Problems, decision problems, computation, and languages: 1.1-1.4

(includes a review of 0.4 and 0.5)

Finite automata and regular expressions:

2.1-2.3 (excluding 2.3.1), 2.6.1-2.6.2, 3.1, 3.2

8.1 (for nonregular languages)

9.1, 9.3 (for regular languages), 10.1.1

Turing machines:

6.1-6.5, 10.1.3, 10.2.3

Context-free grammars and pushdown automata:

4.1, 4.2 (excluding 4.2.5), 4.3, 5.1, 5.2

8.1 (for non-context-free languages)

9.1, 9.2, 9.3 (excluding *CF* substitution)

10.1.2, 10.2.2, 10.3.2

In a quartermester system a restricted version of this coverage could be given. However, it is probably better to concentrate in depth on either finite automata, regular expressions, and Turing machines, or finite automata, regular expressions, context-free grammars, and pushdown automata. It is important to realize that the second approach does not allow for undecidability results to be presented.

If two quartermesters are available, then the first could be devoted to finite automata, regular expressions, Turing machines, and undecidability. The second could then cover context-free grammars, pushdown automata, transductions, and their properties.

Contents

Chapter 7 FUNCTIONS, RELATIONS, AND TRANSLATIONS

Chapter 8 FAMILY RELATIONSHIPS

Chapter 9 CLOSURE PROPERTIES

Part I

INTRODUCTION