

UNDERGRADUATE TEXTS IN COMPUTER SCIENCE

Editors

David Gries

Fred B. Schneider

UNDERGRADUATE TEXTS IN COMPUTER SCIENCE

Beidler, Data Structures and Algorithms

Bergin, Data Structure Programming

Brooks, C Programming: The Essentials for Engineers and Scientists

Brooks, Problem Solving with Fortran 90

Dandamudi, Introduction to Assembly Language Programming

Grillmeyer, Exploring Computer Science with Scheme

Jalote, An Integrated Approach to Software Engineering, Second Edition

Kizza, Ethical and Social Issues in the Information Age

Kozen, Automata and Computability

Merritt and Stix, Migrating from Pascal to C++

Pearce, Programming and Meta-Programming in Scheme

Zeigler, Objects and Systems

Dexter C. Kozen

Automata and Computability

 Springer

Dexter C. Kozen
Department of Computer Science
Cornell University
Ithaca, NY 14853-7501
USA

Series Editors

David Gries
Department of Computer Science
415 Boyd Studies Research Center
The University of Georgia
Athens, Georgia 30602
USA

Fred B. Schneider
Department of Computer Science
Cornell University
4115C Upson Hall
Ithaca, NY 14853-7501
USA

On the cover: Cover photo taken by John Still/Photonica.

With 1 figure.

Library of Congress Cataloging-in-Publication Data
Kozen, Dexter, 1951-

Automata and computability/Dexter C. Kozen.

p. cm. — (Undergraduate texts in computer science)

Includes bibliographical references and index.

ISBN 978-1-4612-7309-7

ISBN 978-1-4612-1844-9 (eBook)

DOI 10.1007/978-1-4612-1844-9

1. Machine theory. 2. Computable functions. I. Title.

II. Series.

QA267.K69 1997

511.3—dc21

96-37409

Printed on acid-free paper.

© 1997 Springer Science+Business Media New York

Originally published by Springer-Verlag New York, Inc. in 1997

Softcover reprint of the hardcover 1st edition 1997

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

9 8 7

ISBN 978-1-4612-7309-7

springeronline.com

To Juris

Preface

These are my lecture notes from CS381/481: Automata and Computability Theory, a one-semester senior-level course I have taught at Cornell University for many years. I took this course myself in the fall of 1974 as a first-year Ph.D. student at Cornell from Juris Hartmanis and have been in love with the subject ever since.

The course is required for computer science majors at Cornell. It exists in two forms: CS481, an honors version; and CS381, a somewhat gentler-paced version. The syllabus is roughly the same, but CS481 goes deeper into the subject, covers more material, and is taught at a more abstract level. Students are encouraged to start off in one or the other, then switch within the first few weeks if they find the other version more suitable to their level of mathematical skill.

The purpose of the course is twofold: to introduce computer science students to the rich heritage of models and abstractions that have arisen over the years; and to develop the capacity to form abstractions of their own and reason in terms of them.

The course is quite mathematical in flavor, and a certain degree of previous mathematical experience is essential for survival. Students should already be conversant with elementary discrete mathematics, including the notions of set, function, relation, product, partial order, equivalence relation, graph, and tree. They should have a repertoire of basic proof techniques at their disposal, including a thorough understanding of the principle of mathematical induction.

The material covered in this text is somewhat more than can be covered in a one-semester course. It is also a mix of elementary and advanced topics. The basic course consists of the lectures numbered 1 through 39. Additionally, I have included several supplementary lectures numbered A through K on various more advanced topics. These can be included or omitted at the instructor's discretion or assigned as extra reading. They appear in roughly the order in which they should be covered.

At first these notes were meant to supplement and not supplant a textbook, but over the years they gradually took on a life of their own. In addition to the notes, I depended on various texts at one time or another: Cutland [30], Harrison [55], Hopcroft and Ullman [60], Lewis and Papadimitriou [79], Machtey and Young [81], and Manna [82]. In particular, the Hopcroft and Ullman text was the standard textbook for the course for many years, and for me it has been an indispensable source of knowledge and insight. All of these texts are excellent references, and I recommend them highly.

In addition to the lectures, I have included 12 homework sets and several miscellaneous exercises. Some of the exercises come with hints and/or solutions; these are indicated by the annotations "H" and "S," respectively. In addition, I have annotated exercises with zero to three stars to indicate relative difficulty.

I have stuck with the format of my previous textbook [72], in which the main text is divided into more or less self-contained lectures, each 4 to 8 pages. Although this format is rather unusual for a textbook, I have found it quite successful. Many readers have commented that they like it because it partitions the subject into bite-sized chunks that can be covered more or less independently.

I owe a supreme debt of gratitude to my wife Frances for her constant love, support, and superhuman patience, especially during the final throes of this project. I am also indebted to the many teachers, colleagues, teaching assistants, and students who over the years have shared the delights of this subject with me and from whom I have learned so much. I would especially like to thank Rick Aaron, Arash Baratloo, Jim Baumgartner, Steve Bloom, Manuel Blum, Amy Briggs, Ashok Chandra, Wilfred Chen, Allan Cheng, Francis Chu, Bob Constable, Devdatt Dubhashi, Peter van Emde Boas, Allen Emerson, András Ferencz, Jeff Foster, Sophia Georgiakaki, David Gries, Joe Halpern, David Harel, Basil Hayek, Tom Henzinger, John Hopcroft, Nick Howe, Doug Ierardi, Tibor Janosi, Jim Jennings, Shyam Kapur, Steve Kautz, Nils Klarlund, Peter Kopke, Vladimir Kotlyar, Alan Kwan, Georges Lauri, Michael Leventon, Jake Levirne, David Liben-Nowell, Yvonne Lo, Steve Mahaney, Nikolay Mateev, Frank McSherry, Albert Meyer, Bob Milnikel, Francesmary Modugno, Anil Nerode, Damian Niwiński, David de la Nuez, Dan Oberlin, Jens Palsberg, Rohit

Parikh, David Pearson, Paul Pedersen, Vaughan Pratt, Zulfikar Ramzan, Jon Rosenberger, Jonathan Rynd, Erik Schmidt, Michael Schwartzbach, Amitabh Shah, Frederick Smith, Kjartan Stefánsson, Colin Stirling, Larry Stockmeyer, Aaron Stump, Jurek Tiuryn, Alex Tsow, Moshe Vardi, Igor Walukiewicz, Rafael Weinstein, Jim Wen, Dan Wineman, Thomas Yan, Paul Zimmons, and many others too numerous to mention. Of course, the greatest of these is Juris Hartmanis, whose boundless enthusiasm for the subject is the ultimate source of my own.

I would be most grateful for suggestions and criticism from readers.

Note added for the third printing. I am indebted to Chris Jeuell for pointing out several typographical errors, which have been corrected in this printing.

Ithaca, New York

Dexter C. Kozen

Contents

Preface	vii
Lectures	1
Introduction	
1 Course Roadmap and Historical Perspective	3
2 Strings and Sets	7
Finite Automata and Regular Sets	
3 Finite Automata and Regular Sets	14
4 More on Regular Sets	19
5 Nondeterministic Finite Automata	25
6 The Subset Construction	32
7 Pattern Matching	40
8 Pattern Matching and Regular Expressions	44
9 Regular Expressions and Finite Automata	49
A Kleene Algebra and Regular Expressions	55
10 Homomorphisms	61
11 Limitations of Finite Automata	67
12 Using the Pumping Lemma	72
13 DFA State Minimization	77
14 A Minimization Algorithm	84
15 Myhill–Nerode Relations	89
16 The Myhill–Nerode Theorem	95

B	Collapsing Nondeterministic Automata	100
C	Automata on Terms	108
D	The Myhill–Nerode Theorem for Term Automata	114
17	Two-Way Finite Automata	119
18	2DFAs and Regular Sets	124
Pushdown Automata and Context-Free Languages		
19	Context-Free Grammars and Languages	129
20	Balanced Parentheses	135
21	Normal Forms	140
22	The Pumping Lemma for CFLs	148
23	Pushdown Automata	157
E	Final State Versus Empty Stack	164
24	PDAs and CFGs	167
25	Simulating NPDAs by CFGs	172
F	Deterministic Pushdown Automata	176
26	Parsing	181
27	The Cocke–Kasami–Younger Algorithm	191
G	The Chomsky–Schützenberger Theorem	198
H	Parikh’s Theorem	201
Turing Machines and Effective Computability		
28	Turing Machines and Effective Computability	206
29	More on Turing Machines	215
30	Equivalent Models	221
31	Universal Machines and Diagonalization	228
32	Decidable and Undecidable Problems	235
33	Reduction	239
34	Rice’s Theorem	245
35	Undecidable Problems About CFLs	249
36	Other Formalisms	256
37	The λ -Calculus	262
I	While Programs	269
J	Beyond Undecidability	274
38	Gödel’s Incompleteness Theorem	282
39	Proof of the Incompleteness Theorem	287
K	Gödel’s Proof	292
Exercises		299
Homework Sets		
	Homework 1	301
	Homework 2	302

Homework 3	303
Homework 4	304
Homework 5	306
Homework 6	307
Homework 7	308
Homework 8	309
Homework 9	310
Homework 10	311
Homework 11	312
Homework 12	313
Miscellaneous Exercises	
Finite Automata and Regular Sets	315
Pushdown Automata and Context-Free Languages	333
Turing Machines and Effective Computability	340
Hints and Solutions	
Hints for Selected Miscellaneous Exercises	351
Solutions to Selected Miscellaneous Exercises	357
References	373
Notation and Abbreviations	381
Index	389