A Second Course in Formal Languages and Automata Theory

Intended for graduate students and advanced undergraduates in computer science, *A Second Course in Formal Languages and Automata Theory* treats topics in the theory of computation not usually covered in a first course.

After a review of basic concepts, the book covers combinatorics on words, regular languages, context-free languages, parsing and recognition, Turing machines, and other language classes. Many topics often absent from other textbooks, such as repetitions in words, state complexity, the interchange lemma, 2DPDAs, and the incompressibility method, are covered here. The author places particular emphasis on the resources needed to represent certain languages. The book also includes a diverse collection of more than 200 exercises, suggestions for term projects, and research problems that remain open.

JEFFREY SHALLIT is professor in the David R. Cheriton School of Computer Science at the University of Waterloo. He is the author of *Algorithmic Number Theory* (co-authored with Eric Bach) and *Automatic Sequences: Theory, Applications, Generalizations* (coauthored with Jean-Paul Allouche). He has published approximately 90 articles on number theory, algebra, automata theory, complexity theory, and the history of mathematics and computing.

A Second Course in Formal Languages and Automata Theory

JEFFREY SHALLIT
University of Waterloo

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo, Delhi

Cambridge University Press
32 Avenue of the Americas, New York, NY 10013-2473, USA

www.cambridge.org
Information on this title: www.cambridge.org/9780521865722

First published 2009

Printed in the United States of America

A catalog record for this publication is available from the British Library.

Library of Congress Cataloging in Publication Data

Shallit, Jeffrey Outlaw.
A second course in formal languages and automata theory / Jeffrey Shallit.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-521-86572-2 (hardback)
1. Formal languages. 2. Machine theory. I. Title.
QA267.3.S53 2009
005.13′1–dc22 2008030065

ISBN 978-0-521-86572-2 hardback

Contents

Preface

Goals of this book

This is a textbook for a second course on formal languages and automata theory.

Many undergraduates in computer science take a course entitled "Introduction to Theory of Computing," in which they learn the basics of finite automata, pushdown automata, context-free grammars, and Turing machines. However, few students pursue advanced topics in these areas, in part because there is no really satisfactory textbook.

For almost 20 years I have been teaching such a second course for fourth-year undergraduate majors and graduate students in computer science at the University of Waterloo: CS 462/662, entitled "Formal Languages and Parsing." For many years we used Hopcroft and Ullman's *Introduction to Automata Theory, Languages, and Computation* as the course text, a book that has proved very influential. (The reader will not have to look far to see its influence on the present book.)

In 2001, however, Hopcroft and Ullman released a second edition of their text that, in the words of one professor, "removed all the good parts." In other words, their second edition is geared toward second- and third-year students, and omits nearly all the advanced topics suitable for fourth-year students and beginning graduate students.

Because the first edition of Hopcroft and Ullman's book is no longer easily available, and because I have been regularly supplementing their book with my own handwritten course notes, it occurred to me that it was a good time to write a textbook on advanced topics in formal languages. The result is this book.

The book contains many topics that are not available in other textbooks. To name just a few, it addresses the Lyndon–Schützenberger theorem, Thue's results on avoiding squares, state complexity of finite automata, Parikh's theorem, the interchange lemma, Earley's parsing method, Kolmogorov complexity, and Cook's theorem on the simulation of 2DPDAs. Furthermore, some well-known

theorems have new (and hopefully simpler) proofs. Finally, there are almost 200 exercises to test students' knowledge. I hope this book will prove useful to advanced undergraduates and beginning graduate students who want to dig a little deeper in the theory of formal languages.

Prerequisites

I assume the reader is familiar with the material contained in a typical first course in the theory of computing and algorithm design. Because not all textbooks use the same terminology and notation, some basic concepts are reviewed in Chapter 1.

Algorithm descriptions

Algorithms in this book are described in a "pseudocode" notation similar to Pascal or C, which should be familiar to most readers. I do not provide a formal definition of this notation. The readers should note that the scope of loops is denoted by indentation.

Proof ideas

Although much of this book follows the traditional theorem/proof style, it does have one nonstandard feature. Many proofs are accompanied by "proof ideas," which attempt to capture the intuition behind the proofs. In some cases, proof ideas are all that is provided.

Common errors

I have tried to point out some common errors that students typically make when encountering this material for the first time.

Exercises

There are a wide variety of exercises, from easy to hard, in no particular order. Most readers will find the exercises with one star challenging and exercises with two stars *very* challenging.

Projects

Each chapter has a small number of suggested projects that are suitable for term papers. Students should regard the provided citations to the literature only as starting points; by tracing forward and backward in the citation history, many more papers can usually be found.

Research problems

Each chapter has a small number of research problems. Currently, no one knows how to solve these problems, so if you make any progress, please contact me.

Acknowledgments

Over the last 18 years, many students in CS 462/662 at the University of Waterloo have contributed to this book by finding errors and suggesting better arguments. I am particularly grateful to the following students (in alphabetical order): Margareta Ackerman, Craig Barkhouse, Matthew Borges, Tracy Damon, Michael DiRamio, Keith Ellul, Sarah Fernandes, Johannes Franz, Cristian Gaspar, Ryan Golbeck, Mike Glover, Daniil Golod, Ryan Hadley, Joe Istead, Richard Kalbfleisch, Jui-Yi Kao, Adam Keanie, Anita Koo, Dalia Krieger, David Landry, Jonathan Lee, Abninder Litt, Brendan Lucier, Bob Lutz, Ian MacDonald, Angela Martin, Glen Martin, Andrew Martinez, Gyanendra Mehta, Olga Miltchman, Siamak Nazari, Sam Ng, Lam Nguyen, Matthew Nichols, Alex Palacios, Joel Reardon, Joe Rideout, Kenneth Rose, Jessica Socha, Douglas Stebila, Wenquan Sun, Aaron Tikuisis, Jim Wallace, Xiang Wang, Zhi Xu, Colin Young, Ning Zhang, and Bryce Zimny.

Four people have been particularly influential in the development of this book, and I single them out for particular thanks: Jonathan Buss, who taught from this book while I was on sabbatical in 2001 and allowed me to use his notes on the closure of CSLs under complement; Narad Rampersad, who taught from this book in 2005, suggested many exercises, and proofread the final draft; Robert Robinson, who taught from a manuscript version of this book at the University of Georgia and identified dozens of errors; and Nic Santean, who carefully read the draft manuscript and made dozens of useful suggestions. All errors that remain, of course, are my responsibility. If you find an error, please send it to me. The current errata list can be found on my homepage; `http://www.cs.uwaterloo.ca/~shallit/` is the URL.

This book was written at the University of Waterloo, Ontario, and the University of Arizona. This work was supported in part by grants from the Natural Sciences and Engineering Research Council (Canada).

Waterloo, Ontario; January 2008 Jeffrey Shallit