Handbook of Mathematical Models for Languages and Computation

Alexander Meduna, Petr Horáček and Martin Tomko

IET PROFESSIONAL APPLICATION OF COMPUTING SERIES 26

Handbook of Mathematical Models for Languages and Computation

Other volumes in this series:

Volume 1 **Knowledge Discovery and Data Mining** M.A. Bramer (Editor)
Volume 3 **Troubled IT Projects: Prevention and turnaround** J.M. Smith
Volume 4 **UML for Systems Engineering: Watching the wheels, 2nd Edition** J. Holt
Volume 5 **Intelligent Distributed Video Surveillance Systems** S.A. Velastin and P. Remagnino (Editors)
Volume 6 **Trusted Computing** C. Mitchell (Editor)
Volume 7 **SysML for Systems Engineering** J. Holt and S. Perry
Volume 8 **Modelling Enterprise Architectures** J. Holt and S. Perry
Volume 9 **Model-Based Requirements Engineering** J. Holt, S. Perry and M. Bownsword
Volume 13 **Trusted Platform Modules: Why, when and how to use them** Ariel Segall
Volume 14 **Foundations for Model-based Systems Engineering: From Patterns to Models** J. Holt, S. Perry and M. Bownsword
Volume 15 **Big Data and Software Defined Networks** J.Taheri (Editor)
Volume 18 **Modeling and Simulation of Complex Communication** M. A. Niazi (Editor)
Volume 20 **SysML for Systems Engineering: A Model-Based Approach, 3rd Edition** J.Holt and S. Perry
Volume 23 **Data as Infrastructure for Smart Cities** L. Suzuki and A. Finkelstein
Volume 24 **Ultrascale Computing Systems** J. Carretero, E. Jeannot and A. Zomaya
Volume 25 **Big Data-Enabled Internet of Things** M. Khan, S. Khan, A. Zomaya (Editors)
Volume 35 **Big Data Recommender Systems Vol 1 & 2** Osman Khalid, Samee U. Khan, Albert Y. Zomaya (Editors)

Handbook of Mathematical Models for Languages and Computation

Alexander Meduna, Petr Horáček and Martin Tomko

The Institution of Engineering and Technology

Published by The Institution of Engineering and Technology, London, United Kingdom

The Institution of Engineering and Technology is registered as a Charity in England & Wales (no. 211014) and Scotland (no. SC038698).

First published 2019

The Institution of Engineering and Technology
Michael Faraday House
Six Hills Way, Stevenage
Herts, SG1 2AY, United Kingdom

www.theiet.org

British Library Cataloguing in Publication Data
A catalogue record for this product is available from the British Library

ISBN 978-1-78561-659-4 (hardback)
ISBN 978-1-78561-660-0 (PDF)

Typeset in India by MPS Limited
Printed in the UK by CPI Group (UK) Ltd, Croydon

To my mother Jarmila, and my father Petr. PH

To Dagmara. AM

To my beloved grandmothers, Marta and Margita,
in memory of my grandfather Ján,
who passed away before I was born,
and in loving memory of my grandfather Emil,
who passed away during my work on this book. MT

Contents

Preface

From a theoretical viewpoint, the present handbook represents a theoretically oriented summary of the knowledge about crucially important mathematical models for languages and computation. It introduces all formalisms concerning these models with enough rigor to make all results quite clear and valid. Every complicated mathematical passage is preceded by its intuitive explanation so that even the most complex parts of the handbook are easy to grasp. Similarly, every new model is preceded by an explanation of its purpose and followed by some examples with comments to reinforce its understanding.

From a practical viewpoint, the handbook pays significant attention to the application and implementation of all the mathematical models. To make quite clear how to encode them, the text contains a large number of computer programs in C#, one of the most popular programming languages at present.

This book is intended for everybody who somehow makes use of mathematical models for languages and computation in their scientific fields. It can also be used as an accompanying text for a class involving these models. Used as a textbook, apart from the theory, it illustrates how to implement and apply the mathematical concepts in computational practice.

This handbook is self-contained in the sense that no other sources are needed to understand the material, so no previous knowledge concerning discrete mathematics is assumed. Nevertheless, a familiarity with the rudiments of high-school mathematics is helpful for a quicker comprehension of the present text. A basic familiarity with a high-level programming language, especially C#, is surely helpful in order to grasp the implementation portion of the text.

The text contains many algorithms. Strictly speaking, every algorithm requires a verification that it terminates and works correctly. However, the termination of the algorithms given in this book is always so obvious that its verification is omitted throughout. The correctness of complicated algorithms is verified. On the other hand, we most often give only the gist of the straightforward algorithms and leave their rigorous verification as an exercise. The text describes the algorithms in Pascal-like notation, which is so simple and intuitive that even the student unfamiliar with the Pascal programming language can immediately pick it up. In this description, a Pascal-like repeat loop is sometimes ended with **until no change**, meaning that the loop is repeated until no change can result from its further repetition. As the clear comprehensibility is of paramount importance in the book, the description of algorithms is often enriched by an explanation in words.

Regarding the technical organization of the text, all the algorithms, examples, conventions, definitions, lemmas, and theorems are sequentially numbered within chapters. Examples and figures are organized similarly.

For further backup materials concerning this handbook, we refer to

http://www.fit. vutbr.cz/~meduna/books/hmmlc.

Synopsis of this handbook

The entire text contains 20 chapters, which are divided into five parts, Parts I–V.

Part I, which consists of Chapters 1–3, reviews all mathematical concepts needed to follow the rest of this handbook. Chapter 1 recalls sets and sequences. It pays special attention to formal languages as sets of finite sequences of symbols because they underlie many central notions in the theory of computation. Chapter 2 examines several concepts concerning relations and functions. Chapter 3 examines a number of concepts from graph theory.

Part II defines classical models for languages and computation based on the mathematical concepts from Part I. Most of these models are underlain by rewriting systems, which are based upon binary relations whose members are called rules. By their rules, these systems repeatedly change sequences of symbols, called strings. They are classified into two categories—generative and accepting language models. Generative models or, briefly, grammars define strings of their languages by generating them from special start symbols. On the other hand, accepting models or, briefly, automata defines the strings of their languages by a rewriting process that starts from these strings and ends in a prescribed set of final strings. Part II consists of Chapters 4–10. Chapter 4 introduces the basic versions of rewriting systems while paying special attention to using them as language-defining devices. Chapter 5 presents finite automata as the simplest versions of automata covered in this book. Chapters 6 discusses generative models called context-free grammars, while Chapter 7 discusses their accepting counterparts—pushdown automata; indeed, both are equally powerful. In Part II, Chapters 8–10 form an inseparable unit, which is crucially important to Part II and, in fact, the book in its entirety. These chapters deal with Turing machines as basic language-defining models for computation. Indeed, based on them, Part II explores the very heart of the foundations of computation. More precisely, Chapter 8 introduces the mathematical notion of a Turing machine, which has become a universally accepted formalization of the intuitive notion of a procedure. Based upon this strictly mathematical notion, Chapters 9 and 10 study the general limits of computation in terms of computability and decidability. Regarding computability, Chapter 9 considers Turing machines as computers of functions over nonnegative integers and demonstrates the existence of functions whose computation cannot be specified by any procedure. As far as decidability is concerned, Chapter 10 formalizes problem-deciding algorithms by Turing machines that halt on every input. It formulates several important problems concerning the language models discussed in this book and constructs algorithms that decide them. On the other hand, Chapter 10 describes several problems that are not decidable by any algorithm. Apart from giving several specific undecidable problems, this book builds up a general theory of undecidability. Finally, the text approaches decidability in a much finer and realistic way. Indeed, it reconsiders problem-deciding algorithms in terms of their computational complexity measured according to time and space requirements. Perhaps most importantly, it shows that although some problems are decidable in principle, they are intractable for absurdly high computational requirements of the algorithms that decide them.

Part III, which consists of Chapters 11–15, covers modern and alternative models for languages and computation. Chapter 11 discusses context-dependent versions of grammatical models. Chapter 12 covers automata and grammars that define languages under various kinds of mathematical regulation. Chapter 13 studies grammatical models that work in parallel. Chapter 14 investigates automata and grammars that work in a discontinuous way. Finally, Chapter 15 defines the automata based upon a generalized versions of pushdown lists.

Part IV, which consists of Chapters 16–19, demonstrates computational applications of mathematical models studied in Parts II and III. Chapter 16 makes many remarks on applications in general. Then, more specifically, Chapters 17 and 18 describe applications in syntax analysis of programming and natural languages, respectively. Chapter 19 shows applications in computational biology.

Part V, which consists of a single chapter—Chapter 20, closes the entire book by adding several important remarks concerning its coverage. It sums up all the coverage contained in the text. It also sketches important current investigation trends. Finally, Chapter 20 makes several bibliographical remarks.

Acknowledgements

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science – LQ1602; the TAČR grant TE01020415; and the BUT grant FIT-S-17-3964.

Handbook of Mathematical Models for Languages and Computation

The theory of computation is used to address challenges arising in many computer science areas such as artificial intelligence, language processors, compiler writing, information and coding systems, programming language design, computer architecture and more. To grasp topics concerning this theory readers need to familiarize themselves with its computational and language models, based on concepts of discrete mathematics including sets, relations, functions, graphs and logic.

This handbook introduces with rigor the important concepts of this kind and uses them to cover the most important mathematical models for languages and computation, such as various classical as well as modern automata and grammars. It explains their use in such crucially significant topics of computation theory as computability, decidability, and computational complexity. The authors pay special attention to the implementation of all these mathematical concepts and models and explains clearly how to encode them in computational practice. All computer programs are written in C#.

About the Authors

Alexander Meduna is Full Professor of Computer Science at the Brno University of Technology, Czech Republic. He has taught mathematics and computer science at various European, Asian, and American universities, including the University of Missouri, USA, where he spent a decade teaching advanced topics of the formal language theory and its applications in computer science. He is the author of several books and many papers on the topic.

Petr Horáček works as a software engineer in research and development for BioVendor Instruments a.s. (a laboratory for automation and diagnostic devices) in Brno, Czech Republic. Under the supervision of Alexander Meduna, he received his PhD from the Brno University of Technology, Czech Republic, in 2015.

Martin Tomko is a PhD Student supervised by Alexander Meduna at the Faculty of Information Technology, Brno University of Technology, Czech Republic.

ISBN 978-1-78561-659-4

9 781785 616594 >

Computing

The Institution of Engineering and Technology • www.theiet.org
978-1-78561-659-4