

11755

Handbook of Mathematical Models for Languages and Computation

Alexander Meduna, Petr Horáček and Martin Tomko



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

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

The Institution of Engineering and Technology

Fakulta informačních technologií
Krištofůva

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).

© The Institution of Engineering and Technology 2020

First published 2019

This publication is copyright under the Berne Convention and the Universal Copyright Convention. All rights reserved. Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may be reproduced, stored or transmitted, in any form or by any means, only with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publisher at the undermentioned address:

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

www.theiet.org

While the authors and publisher believe that the information and guidance given in this work are correct, all parties must rely upon their own skill and judgement when making use of them. Neither the authors nor publisher assumes any liability to anyone for any loss or damage caused by any error or omission in the work, whether such an error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

The moral rights of the authors to be identified as authors of this work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

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)

Knihovna FIT



4338126834

Typeset in India by MPS Limited

Printed in the UK by CPI Group (UK) Ltd, Croydon

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

Knihovna

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	xiii
Acknowledgements	xvii
List of implementations	xix
List of symbols	xxi
List of mathematical models	xxiii
List of language families	xxv
PART I Basic mathematical concepts	1
1 Sets, sequences, and languages	3
1.1 Sets	3
1.2 Sequences	10
1.3 Formal languages	11
2 Relations and functions	21
2.1 Relations	21
2.2 Functions	25
3 Graphs	29
3.1 Directed graphs	29
3.2 Trees	30
PART II Classical models for languages and computation	35
4 Relations and language models	37
4.1 Rewriting systems	37
4.2 Language-defining models	42
5 Finite automata	47
5.1 Mathematical elements of finite automata	47
5.1.1 How to specify finite automata	48
5.2 Finite automata that always read	53
5.3 Determinism	60
5.4 Reduction and minimization	71
5.5 Regular expressions	75

6	Context-free grammars	87
6.1	Mathematical elements of context-free grammars	87
6.2	Canonical derivations and derivation trees	94
6.2.1	Leftmost derivations	94
6.2.2	Rightmost derivations	96
6.2.3	Derivation trees	96
6.2.4	Ambiguity	98
6.3	Useless symbols and their elimination	102
6.4	Erasing rules and their elimination	109
6.5	Single rules and their elimination	114
6.6	Chomsky normal form	118
6.7	Left recursion and its removal	123
6.7.1	Direct left recursion and its elimination	124
6.7.2	Left recursion and its elimination	125
6.7.3	Right recursion	132
6.8	Greibach normal form	132
7	Pushdown automata	139
7.1	Pushdown automata	139
7.2	Pushdown automata and context-free grammars are equivalent	146
7.3	Three types of acceptance by pushdown automata	150
7.4	Determinism	152
8	Turing machines	159
8.1	Turing machines and their languages	159
8.2	Determinism	167
8.3	Universality	171
8.3.1	Turing machine codes	171
8.3.2	Construction of universal Turing machines	176
9	Computability	179
9.1	Functions computed by Turing machines	179
9.2	Mathematical theory of computability: an introduction	184
10	Decidability	191
10.1	Turing deciders	191
10.2	Decidable problems	194
10.2.1	Decidable problems for finite automata	194
10.2.2	Decidable problems for context-free grammars	196
10.3	Undecidability: diagonalization	198
10.4	Undecidability: reduction	201
10.5	Undecidability: a general approach to reduction	204
10.6	Computational complexity	208
10.6.1	Time complexity	209
10.6.2	Space complexity	211

PART III Alternative models for languages and computation	213
11 Context-dependent grammars	215
11.1 Tightly context-dependent grammars	215
11.2 Loosely context-dependent grammars	232
12 Regulated models	293
12.1 Grammars regulated by states	293
12.1.1 Definitions and examples	293
12.1.2 Generative power	295
12.2 Grammars regulated by control languages	295
12.2.1 Definitions and examples	295
12.2.2 Generative power	299
12.3 Matrix grammars	299
12.3.1 Definitions and examples	299
12.3.2 Generative power	301
12.3.3 Even matrix grammars	301
12.4 Programmed grammars	313
12.4.1 Definitions and examples	313
12.4.2 Generative power	316
12.5 Regulated automata and computation	316
12.5.1 Self-regulating automata	316
12.5.2 Self-regulating finite automata	317
12.5.3 Self-regulating pushdown automata	332
12.5.4 Open problems	335
12.5.5 Regulated acceptance with control languages	335
12.5.6 Finite automata regulated by control languages	335
12.5.7 Pushdown automata regulated by control languages	349
12.5.8 Self-reproducing pushdown transducers	360
12.5.9 Definitions	361
12.5.10 Results	362
13 Parallel grammatical models	367
13.1 Partially parallel grammars	367
13.1.1 Definitions and examples	367
13.1.2 Generative power	371
13.1.3 Normal forms	371
13.1.4 Reduction	373
13.1.5 Economical transformations	385
13.2 Totally parallel grammars	392
13.2.1 Context-conditional ETOL grammars	393
13.2.2 Forbidding ETOL grammars	400
13.2.3 Simple semi-conditional ETOL grammars	419
13.2.4 Left random context ETOL grammars	433
13.3 Multigenerative grammar systems and parallel computation	449
13.3.1 Multigenerative grammar systems	450
13.3.2 Leftmost multigenerative grammar systems	466

14	Jumping models	479
14.1	Sequential jumping grammars	479
14.2	Parallel jumping grammars	494
14.2.1	Definitions	496
14.2.2	Results	498
14.3	Jumping automata	524
14.3.1	Definitions and examples	526
14.3.2	Properties	527
15	Deep pushdown automata	545
15.1	Basic model	545
15.1.1	Definitions and examples	546
15.1.2	Accepting power	547
15.1.3	Open problems	555
15.2	Restricted versions	556
15.2.1	Preliminaries and definitions	557
15.2.2	Results	557
15.2.3	Open problems	562
PART IV	Applications	563
16	Applications in general	565
16.1	Applications in computational linguistics: general comments	565
16.2	Applications in computational biology: general comments	567
17	Applications in syntax analysis: programming languages	569
17.1	General parsers	570
17.1.1	Syntax specified by context-free grammars	571
17.1.2	Top-down parsing	573
17.1.3	Bottom-up parsing	576
17.2	Top-down parsers	582
17.2.1	Predictive sets and LL grammars	582
17.2.2	LL grammars	589
17.2.3	Predictive parsing	591
17.2.4	Predictive recursive-descent parsing	591
17.2.5	Predictive table-driven parsing	597
17.2.6	Handling errors	603
17.2.7	Exclusion of left recursion	605
17.3	Bottom-up parsers	605
17.3.1	Operator-precedence parsing	606
17.3.2	Operator-precedence parser	606
17.3.3	Construction of operator-precedence parsing table	614
17.3.4	Handling errors	615
17.3.5	Operator-precedence parsers for other expressions	619
17.3.6	LR parsing	621
17.3.7	LR parsing algorithm	621

17.3.8 Construction of the LR table	630
17.3.9 Handling errors in LR parsing	643
17.4 Syntax-directed translation: an implementation	646
18 Applications in syntax analysis: natural languages	661
18.1 Syntax and related linguistic terminology	661
18.1.1 Introduction by way of examples	661
18.1.2 Terminology	663
18.1.3 Verbs	664
18.1.4 Personal pronouns	664
18.2 Transformational scattered context grammars	666
18.3 Scattered context in English syntax	668
18.3.1 Clauses with <i>neither</i> and <i>nor</i>	669
18.3.2 Existential clauses	670
18.3.3 Interrogative clauses	671
18.3.4 Question tags	673
18.3.5 Generation of grammatical sentences	675
19 Applications in biology	679
19.1 Applications	679
19.1.1 Death	682
19.1.2 Degeneration	683
19.2 Implementation	686
PART V Conclusion	693
20 Concluding remarks	695
20.1 Summary of the book	695
20.2 Latest trends and open problems	698
20.3 Bibliographical remarks	700
20.3.1 Grammars	700
20.3.2 Automata	701
References	702
Index	721

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 define 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. Chapter 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.



4338126834

Handbook of Mathematical Models 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.

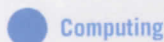
Petr Horáček works as a software engineer in research and development for BioVendor Instruments a.s. (a laboratory for automation and diagnostic devices) in Brno, Czech Republic. Under the supervision of Alexander Meduna, he received his PhD from the Brno University of Technology, Czech Republic, in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