

Theory of Computer Science

Automata, Languages and Computation

Third Edition

K.L.P. Mishra
N. Chandrasekaran



THEORY OF COMPUTER SCIENCE

Automata, Languages and Computation

THIRD EDITION

K.L.P. MISHRA

Formerly Professor

*Department of Electrical and Electronics Engineering
and Principal, Regional Engineering College
Tiruchirapalli*

N. CHANDRASEKARAN

Professor

*Department of Mathematics
St. Joseph's College
Tiruchirapalli*

PHI Learning Private Limited

Delhi-110092
2016

₹ 275.00

THEORY OF COMPUTER SCIENCE (Automata, Languages and Computation), 3rd ed.

K.L.P. Mishra and N. Chandrasekaran

© 2007 by PHI Learning Private Limited, Delhi. All rights reserved. No part of this book may be reproduced in any form, by mimeograph or any other means, without permission in writing from the publisher.

ISBN-978-81-203-2968-3

The export rights of this book are vested solely with the publisher.

Thirty-fourth Printing (Third Edition)

...

...

April, 2016

Published by Asoke K. Ghosh, PHI Learning Private Limited, Rimjhim House, 111, Patparganj Industrial Estate, Delhi-110092 and Printed by Rajkamal Electric Press, Plot No. 2, Phase IV, HSIDC, Kundli-131028, Sonapat, Haryana.



Contents

<i>Preface</i>	<i>ix</i>
<i>Notations</i>	<i>xi</i>

1. PROPOSITIONS AND PREDICATES **1–35**

1.1	Propositions (or Statements)	<i>1</i>
1.1.1	Connectives (Propositional Connectives or Logical Connectives)	<i>2</i>
1.1.2	Well-formed Formulas	<i>6</i>
1.1.3	Truth Table for a Well-formed Formula	<i>7</i>
1.1.4	Equivalence of Well-formed Formulas	<i>9</i>
1.1.5	Logical Identities	<i>9</i>
1.2	Normal Forms of Well-formed Formulas	<i>11</i>
1.2.1	Construction to Obtain a Disjunctive Normal Form of a Given Formula	<i>11</i>
1.2.2	Construction to Obtain the Principal Disjunctive Normal Form of a Given Formula	<i>12</i>
1.3	Rules of Inference for Propositional Calculus (Statement Calculus)	<i>15</i>
1.4	Predicate Calculus	<i>19</i>
1.4.1	Predicates	<i>19</i>
1.4.2	Well-formed Formulas of Predicate Calculus	<i>21</i>
1.5	Rules of Inference for Predicate Calculus	<i>23</i>
1.6	Supplementary Examples	<i>26</i>
	<i>Self-Test</i>	<i>31</i>
	<i>Exercises</i>	<i>32</i>

2. MATHEMATICAL PRELIMINARIES**36–70**

2.1	Sets, Relations and Functions	36
2.1.1	Sets and Subsets	36
2.1.2	Sets with One Binary Operation	37
2.1.3	Sets with Two Binary Operations	39
2.1.4	Relations	40
2.1.5	Closure of Relations	43
2.1.6	Functions	45
2.2	Graphs and Trees	47
2.2.1	Graphs	47
2.2.2	Trees	49
2.3	Strings and Their Properties	54
2.3.1	Operations on Strings	54
2.3.2	Terminal and Nonterminal Symbols	56
2.4	Principle of Induction	57
2.4.1	Method of Proof by Induction	57
2.4.2	Modified Method of Induction	58
2.4.3	Simultaneous Induction	60
2.5	Proof by Contradiction	61
2.6	Supplementary Examples	62
	<i>Self-Test</i>	66
	<i>Exercises</i>	67

3. THE THEORY OF AUTOMATA**71–106**

3.1	Definition of an Automaton	71
3.2	Description of a Finite Automaton	73
3.3	Transition Systems	74
3.4	Properties of Transition Functions	75
3.5	Acceptability of a String by a Finite Automaton	77
3.6	Nondeterministic Finite State Machines	78
3.7	The Equivalence of DFA and N DFA	80
3.8	Mealy and Moore Models	84
3.8.1	Finite Automata with Outputs	84
3.8.2	Procedure for Transforming a Mealy Machine into a Moore Machine	85
3.8.3	Procedure for Transforming a Moore Machine into a Mealy Machine	87
3.9	Minimization of Finite Automata	91
3.9.1	Construction of Minimum Automaton	92
3.10	Supplementary Examples	97
	<i>Self-Test</i>	103
	<i>Exercises</i>	104

4. FORMAL LANGUAGES	107–135
4.1 Basic Definitions and Examples	107
4.1.1 Definition of a Grammar	109
4.1.2 Derivations and the Language Generated by a Grammar	110
4.2 Chomsky Classification of Languages	120
4.3 Languages and Their Relation	123
4.4 Recursive and Recursively Enumerable Sets	124
4.5 Operations on Languages	126
4.6 Languages and Automata	128
4.7 Supplementary Examples	129
<i>Self-Test</i>	132
<i>Exercises</i>	134
5. REGULAR SETS AND REGULAR GRAMMARS	136–179
5.1 Regular Expressions	136
5.1.1 Identities for Regular Expressions	138
5.2 Finite Automata and Regular Expressions	140
5.2.1 Transition System Containing Λ -moves	140
5.2.2 NDFAs with Λ -moves and Regular Expressions	142
5.2.3 Conversion of Nondeterministic Systems to Deterministic Systems	146
5.2.4 Algebraic Method Using Arden's Theorem	148
5.2.5 Construction of Finite Automata Equivalent to a Regular Expression	153
5.2.6 Equivalence of Two Finite Automata	157
5.2.7 Equivalence of Two Regular Expressions	160
5.3 Pumping Lemma for Regular Sets	162
5.4 Application of Pumping Lemma	163
5.5 Closure Properties of Regular Sets	165
5.6 Regular Sets and Regular Grammars	167
5.6.1 Construction of a Regular Grammar Generating $T(M)$ for a Given DFA M	168
5.6.2 Construction of a Transition System M Accepting $L(G)$ for a Given Regular Grammar G	169
5.7 Supplementary Examples	170
<i>Self-Test</i>	175
<i>Exercises</i>	176
6. CONTEXT-FREE LANGUAGES	180–226
6.1 Context-free Languages and Derivation Trees	180
6.1.1 Derivation Trees	181
6.2 Ambiguity in Context-free Grammars	188

6.3	Simplification of Context-free Grammars	189
6.3.1	Construction of Reduced Grammars	190
6.3.2	Elimination of Null Productions	196
6.3.3	Elimination of Unit Productions	199
6.4	Normal Forms for Context-free Grammars	201
6.4.1	Chomsky Normal Form	201
6.4.2	Greibach Normal Form	206
6.5	Pumping Lemma for Context-free Languages	213
6.6	Decision Algorithms for Context-free Languages	217
6.7	Supplementary Examples	218
	<i>Self-Test</i>	223
	<i>Exercises</i>	224
7.	PUSHDOWN AUTOMATA	227–266
7.1	Basic Definitions	227
7.2	Acceptance by pda	233
7.3	Pushdown Automata and Context-free Languages	240
7.4	Parsing and Pushdown Automata	251
7.4.1	Top-down Parsing	252
7.4.2	Top-down Parsing Using Deterministic pda's	256
7.4.3	Bottom-up Parsing	258
7.5	Supplementary Examples	260
	<i>Self-Test</i>	264
	<i>Exercises</i>	265
8.	LR(k) GRAMMARS	267–276
8.1	LR(k) Grammars	267
8.2	Properties of LR(k) Grammars	270
8.3	Closure Properties of Languages	272
8.4	Supplementary Examples	272
	<i>Self-Test</i>	273
	<i>Exercises</i>	274
9.	TURING MACHINES AND LINEAR BOUNDED AUTOMATA	277–308
9.1	Turing Machine Model	278
9.2	Representation of Turing Machines	279
9.2.1	Representation by Instantaneous Descriptions	279
9.2.2	Representation by Transition Table	280
9.2.3	Representation by Transition Diagram	281
9.3	Language Acceptability by Turing Machines	283
9.4	Design of Turing Machines	284
9.5	Description of Turing Machines	289

9.6	Techniques for TM Construction	289	
9.6.1	Turing Machine with Stationary Head	289	
9.6.2	Storage in the State	290	
9.6.3	Multiple Track Turing Machine	290	
9.6.4	Subroutines	290	
9.7	Variants of Turing Machines	292	
9.7.1	Multitape Turing Machines	292	
9.7.2	Nondeterministic Turing Machines	295	
9.8	The Model of Linear Bounded Automaton	297	
9.8.1	Relation Between LBA and Context-sensitive Languages	299	
9.9	Turing Machines and Type 0 Grammars	299	
9.9.1	Construction of a Grammar Corresponding to TM	299	
9.10	Linear Bounded Automata and Languages	301	
9.11	Supplementary Examples	303	
	<i>Self-Test</i>	307	
	<i>Exercises</i>	308	
10.	DECIDABILITY AND RECURSIVELY ENUMERABLE LANGUAGES		309–321
10.1	The Definition of an Algorithm	309	
10.2	Decidability	310	
10.3	Decidable Languages	311	
10.4	Undecidable Languages	313	
10.5	Halting Problem of Turing Machine	314	
10.6	The Post Correspondence Problem	315	
10.7	Supplementary Examples	317	
	<i>Self-Test</i>	319	
	<i>Exercises</i>	319	
11.	COMPUTABILITY		322–345
11.1	Introduction and Basic Concepts	322	
11.2	Primitive Recursive Functions	323	
11.2.1	Initial Functions	323	
11.2.2	Primitive Recursive Functions Over N	325	
11.2.3	Primitive Recursive Functions Over $\{a, b\}$	327	
11.3	Recursive Functions	329	
11.4	Partial Recursive Functions and Turing Machines	332	
11.4.1	Computability	332	
11.4.2	A Turing Model for Computation	333	
11.4.3	Turing-computable Functions	333	
11.4.4	Construction of the Turing Machine That Can Compute the Zero Function Z	334	
11.4.5	Construction of the Turing Machine for Computing—The Successor Function	335	

11.4.6	Construction of the Turing Machine for Computing the Projection U_i^m	336
11.4.7	Construction of the Turing Machine That Can Perform Composition	338
11.4.8	Construction of the Turing Machine That Can Perform Recursion	339
11.4.9	Construction of the Turing Machine That Can Perform Minimization	340
11.5	Supplementary Examples	340
	<i>Self-Test</i>	342
	<i>Exercises</i>	343
12.	COMPLEXITY	346–371
12.1	Growth Rate of Functions	346
12.2	The Classes P and NP	349
12.3	Polynomial Time Reduction and <i>NP</i> -completeness	351
12.4	Importance of <i>NP</i> -complete Problems	352
12.5	SAT is <i>NP</i> -complete	353
	12.5.1 Boolean Expressions	353
	12.5.2 Coding a Boolean Expression	353
	12.5.3 Cook's Theorem	354
12.6	Other <i>NP</i> -complete Problems	359
12.7	Use of <i>NP</i> -completeness	360
12.8	Quantum Computation	360
	12.8.1 Quantum Computers	361
	12.8.2 Church–Turing Thesis	362
	12.8.3 Power of Quantum Computation	363
	12.8.4 Conclusion	364
12.9	Supplementary Examples	365
	<i>Self-Test</i>	369
	<i>Exercises</i>	370
	<i>Answers to Self-Tests</i>	373–374
	<i>Solutions (or Hints) to Chapter-end Exercises</i>	375–415
	<i>Further Reading</i>	417–418
	<i>Index</i>	419–422



Preface

The enlarged third edition of *Theory of Computer Science* is the result of the enthusiastic reception given to earlier editions of this book and the feedback received from the students and teachers who used the second edition for several years.

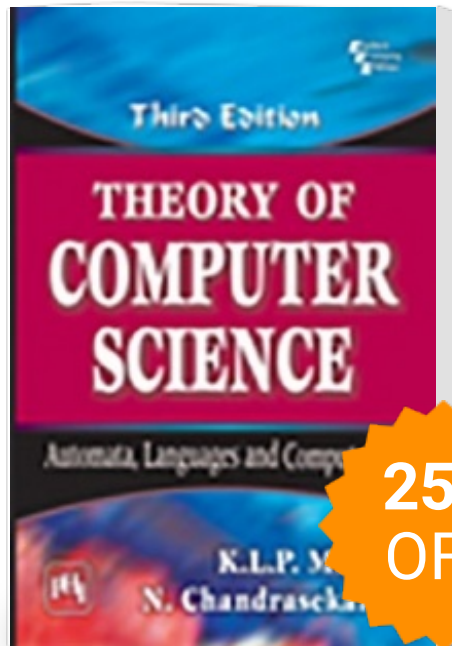
The new edition deals with all aspects of theoretical computer science, namely **automata**, **formal languages**, **computability** and **complexity**. Very few books combine all these theories and give adequate examples. This book provides numerous examples that illustrate the basic concepts. It is profusely illustrated with diagrams. While dealing with theorems and algorithms, the emphasis is on constructions. Each construction is immediately followed by an example and only then the formal proof is given so that the student can master the technique involved in the construction before taking up the formal proof.

The key feature of the book that sets it apart from other books is the provision of detailed solutions (at the end of the book) to chapter-end exercises.

The chapter on Propositions and Predicates (Chapter 10 of the second edition) is now the first chapter in the new edition. The changes in other chapters have been made without affecting the structure of the second edition. The chapter on Turing machines (Chapter 7 of the second edition) has undergone major changes.

A novel feature of the third edition is the addition of objective type questions in each chapter under the heading Self-Test. This provides an opportunity to the student to test whether he has fully grasped the fundamental concepts. Besides, a total number of 83 additional solved examples have been added as Supplementary Examples which enhance the variety of problems dealt with in the book.

Theory Of Computer Science : Automata, Languages And Computation



Publisher : PHI Learning

ISBN : 9788120329683

Author : MISHRA,
K.L.P., CHANDRASEKARAN,
N.

Type the URL : <http://www.kopykitab.com/product/7411>



Get this eBook