

# CONTENTS

<b>Preface to the First Edition</b>	<b>xi</b>
To the student . . . . .	xi
To the educator . . . . .	xii
The first edition . . . . .	xiii
Feedback to the author . . . . .	xiii
Acknowledgments . . . . .	xiv

<b>Preface to the Second Edition</b>	<b>xvii</b>
--------------------------------------	-------------

<b>0 Introduction</b>	<b>1</b>
0.1 Automata, Computability, and Complexity . . . . .	1
Complexity theory . . . . .	2
Computability theory . . . . .	2
Automata theory . . . . .	3
0.2 Mathematical Notions and Terminology . . . . .	3
Sets . . . . .	3
Sequences and tuples . . . . .	6
Functions and relations . . . . .	7
Graphs . . . . .	10
Strings and languages . . . . .	13
Boolean logic . . . . .	14
Summary of mathematical terms . . . . .	16
0.3 Definitions, Theorems, and Proofs . . . . .	17
Finding proofs . . . . .	17
0.4 Types of Proof . . . . .	21
Proof by construction . . . . .	21
Proof by contradiction . . . . .	21
Proof by induction . . . . .	22
<i>Exercises, Problems, and Solutions</i> . . . . .	25

**Part One: Automata and Languages 29**

<b>1</b>	<b>Regular Languages</b>	<b>31</b>
1.1	Finite Automata . . . . .	31
	Formal definition of a finite automaton . . . . .	35
	Examples of finite automata . . . . .	37
	Formal definition of computation . . . . .	40
	Designing finite automata . . . . .	41
	The regular operations . . . . .	44
1.2	Nondeterminism . . . . .	47
	Formal definition of a nondeterministic finite automaton . . . . .	53
	Equivalence of NFAs and DFAs . . . . .	54
	Closure under the regular operations . . . . .	58
1.3	Regular Expressions . . . . .	63
	Formal definition of a regular expression . . . . .	64
	Equivalence with finite automata . . . . .	66
1.4	Nonregular Languages . . . . .	77
	The pumping lemma for regular languages . . . . .	77
	<i>Exercises, Problems, and Solutions</i> . . . . .	82
<b>2</b>	<b>Context-Free Languages</b>	<b>99</b>
2.1	Context-free Grammars . . . . .	100
	Formal definition of a context-free grammar . . . . .	102
	Examples of context-free grammars . . . . .	103
	Designing context-free grammars . . . . .	104
	Ambiguity . . . . .	105
	Chomsky normal form . . . . .	106
2.2	Pushdown Automata . . . . .	109
	Formal definition of a pushdown automaton . . . . .	111
	Examples of pushdown automata . . . . .	112
	Equivalence with context-free grammars . . . . .	115
2.3	Non-context-free Languages . . . . .	123
	The pumping lemma for context-free languages . . . . .	123
	<i>Exercises, Problems, and Solutions</i> . . . . .	128

**Part Two: Computability Theory 135**

<b>3</b>	<b>The Church–Turing Thesis</b>	<b>137</b>
3.1	Turing Machines . . . . .	137
	Formal definition of a Turing machine . . . . .	139
	Examples of Turing machines . . . . .	142
3.2	Variants of Turing Machines . . . . .	148
	Multitape Turing machines . . . . .	148
	Nondeterministic Turing machines . . . . .	150
	Enumerators . . . . .	152

Equivalence with other models . . . . .	153
3.3 The Definition of Algorithm . . . . .	154
Hilbert's problems . . . . .	154
Terminology for describing Turing machines . . . . .	156
<i>Exercises, Problems, and Solutions</i> . . . . .	159
<b>4 Decidability</b>	<b>165</b>
4.1 Decidable Languages . . . . .	166
Decidable problems concerning regular languages . . . . .	166
Decidable problems concerning context-free languages . . . . .	170
4.2 The Halting Problem . . . . .	173
The diagonalization method . . . . .	174
The halting problem is undecidable . . . . .	179
A Turing-unrecognizable language . . . . .	181
<i>Exercises, Problems, and Solutions</i> . . . . .	182
<b>5 Reducibility</b>	<b>187</b>
5.1 Undecidable Problems from Language Theory . . . . .	188
Reductions via computation histories . . . . .	192
5.2 A Simple Undecidable Problem . . . . .	199
5.3 Mapping Reducibility . . . . .	206
Computable functions . . . . .	206
Formal definition of mapping reducibility . . . . .	207
<i>Exercises, Problems, and Solutions</i> . . . . .	211
<b>6 Advanced Topics in Computability Theory</b>	<b>217</b>
6.1 The Recursion Theorem . . . . .	217
Self-reference . . . . .	218
Terminology for the recursion theorem . . . . .	221
Applications . . . . .	222
6.2 Decidability of logical theories . . . . .	224
A decidable theory . . . . .	227
An undecidable theory . . . . .	229
6.3 Turing Reducibility . . . . .	232
6.4 A Definition of Information . . . . .	233
Minimal length descriptions . . . . .	234
Optimality of the definition . . . . .	238
Incompressible strings and randomness . . . . .	239
<i>Exercises, Problems, and Solutions</i> . . . . .	242
 <b>Part Three: Complexity Theory</b>	 <b>245</b>
<b>7 Time Complexity</b>	<b>247</b>
7.1 Measuring Complexity . . . . .	247
Big- <i>O</i> and small- <i>o</i> notation . . . . .	248

	Analyzing algorithms . . . . .	251
	Complexity relationships among models . . . . .	254
7.2	The Class P . . . . .	256
	Polynomial time . . . . .	256
	Examples of problems in P . . . . .	258
7.3	The Class NP . . . . .	264
	Examples of problems in NP . . . . .	267
	The P versus NP question . . . . .	269
7.4	NP-completeness . . . . .	271
	Polynomial time reducibility . . . . .	272
	Definition of NP-completeness . . . . .	276
	The Cook–Levin Theorem . . . . .	276
7.5	Additional NP-complete Problems . . . . .	283
	The vertex cover problem . . . . .	284
	The Hamiltonian path problem . . . . .	286
	The subset sum problem . . . . .	291
	<i>Exercises, Problems, and Solutions</i> . . . . .	294
<b>8</b>	<b>Space Complexity</b>	<b>303</b>
8.1	Savitch’s Theorem . . . . .	305
8.2	The Class PSPACE . . . . .	308
8.3	PSPACE-completeness . . . . .	309
	The TQBF problem . . . . .	310
	Winning strategies for games . . . . .	313
	Generalized geography . . . . .	315
8.4	The Classes L and NL . . . . .	320
8.5	NL-completeness . . . . .	323
	Searching in graphs . . . . .	325
8.6	NL equals coNL . . . . .	326
	<i>Exercises, Problems, and Solutions</i> . . . . .	328
<b>9</b>	<b>Intractability</b>	<b>335</b>
9.1	Hierarchy Theorems . . . . .	336
	Exponential space completeness . . . . .	343
9.2	Relativization . . . . .	348
	Limits of the diagonalization method . . . . .	349
9.3	Circuit Complexity . . . . .	351
	<i>Exercises, Problems, and Solutions</i> . . . . .	360
<b>10</b>	<b>Advanced topics in complexity theory</b>	<b>365</b>
10.1	Approximation Algorithms . . . . .	365
10.2	Probabilistic Algorithms . . . . .	368
	The class BPP . . . . .	368
	Primality . . . . .	371
	Read-once branching programs . . . . .	376
10.3	Alternation . . . . .	380

Alternating time and space . . . . .	381
The Polynomial time hierarchy . . . . .	386
10.4 Interactive Proof Systems . . . . .	387
Graph nonisomorphism . . . . .	387
Definition of the model . . . . .	388
IP = PSPACE . . . . .	390
10.5 Parallel Computation . . . . .	399
Uniform Boolean circuits . . . . .	400
The class NC . . . . .	402
P-completeness . . . . .	404
10.6 Cryptography . . . . .	405
Secret keys . . . . .	405
Public-key cryptosystems . . . . .	407
One-way functions . . . . .	407
Trapdoor functions . . . . .	409
<i>Exercises, Problems, and Solutions</i> . . . . .	411
<b>Selected Bibliography</b>	<b>415</b>
<b>Index</b>	<b>421</b>