

# CONTENTS

<b>Preface</b>	<b>xi</b>
To the student . . . . .	xi
To the educator . . . . .	xii
The current edition . . . . .	xiii
Feedback to the author . . . . .	xiii
Acknowledgments . . . . .	xiv
<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 . . . . .	23
<i>Exercises and Problems</i> . . . . .	25
<b>Part One: Automata and Languages</b>	<b>29</b>
<b>1 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 and Problems</i> . . . . .	83
<b>2</b>	<b>Context-Free Languages</b>	<b>91</b>
2.1	Context-free Grammars . . . . .	92
	Formal definition of a context-free grammar . . . . .	94
	Examples of context-free grammars . . . . .	95
	Designing context-free grammars . . . . .	96
	Ambiguity . . . . .	97
	Chomsky normal form . . . . .	98
2.2	Pushdown Automata . . . . .	101
	Formal definition of a pushdown automaton . . . . .	103
	Examples of pushdown automata . . . . .	104
	Equivalence with context-free grammars . . . . .	106
2.3	Non-context-free Languages . . . . .	115
	The pumping lemma for context-free languages . . . . .	115
	<i>Exercises and Problems</i> . . . . .	119
<b>Part Two: Computability Theory</b>		<b>123</b>
<b>3</b>	<b>The Church–Turing Thesis</b>	<b>125</b>
3.1	Turing Machines . . . . .	125
	Formal definition of a Turing machine . . . . .	127
	Examples of Turing machines . . . . .	130
3.2	Variants of Turing Machines . . . . .	136
	Multitape Turing machines . . . . .	136
	Nondeterministic Turing machines . . . . .	138
	Enumerators . . . . .	140
	Equivalence with other models . . . . .	141
3.3	The Definition of Algorithm . . . . .	142
	Hilbert’s problems . . . . .	142
	Terminology for describing Turing machines . . . . .	144
	<i>Exercises and Problems</i> . . . . .	147

<b>4</b>	<b>Decidability</b>	<b>151</b>
4.1	Decidable Languages . . . . .	152
	Decidable problems concerning regular languages . . . . .	152
	Decidable problems concerning context-free languages . . . . .	156
4.2	The Halting Problem . . . . .	159
	The diagonalization method . . . . .	160
	The halting problem is undecidable . . . . .	165
	A Turing-unrecognizable language . . . . .	167
	<i>Exercises and Problems</i> . . . . .	168
<b>5</b>	<b>Reducibility</b>	<b>171</b>
5.1	Undecidable Problems from Language Theory . . . . .	172
	Reductions via computation histories . . . . .	176
5.2	A Simple Undecidable Problem . . . . .	183
5.3	Mapping Reducibility . . . . .	189
	Computable functions . . . . .	190
	Formal definition of mapping reducibility . . . . .	191
	<i>Exercises and Problems</i> . . . . .	195
<b>6</b>	<b>Advanced Topics in Computability Theory</b>	<b>197</b>
6.1	The Recursion Theorem . . . . .	197
	Self-reference . . . . .	198
	Terminology for the recursion theorem . . . . .	201
	Applications . . . . .	202
6.2	Decidability of logical theories . . . . .	204
	A decidable theory . . . . .	206
	An undecidable theory . . . . .	209
6.3	Turing Reducibility . . . . .	211
6.4	A Definition of Information . . . . .	213
	Minimal length descriptions . . . . .	214
	Optimality of the definition . . . . .	217
	Incompressible strings and randomness . . . . .	217
	<i>Exercises and Problems</i> . . . . .	220

## **Part Three: Complexity Theory** **223**

<b>7</b>	<b>Time Complexity</b>	<b>225</b>
7.1	Measuring Complexity . . . . .	225
	Big- <i>O</i> and small- <i>o</i> notation . . . . .	226
	Analyzing algorithms . . . . .	229
	Complexity relationships among models . . . . .	231
7.2	The Class P . . . . .	234
	Polynomial time . . . . .	234
	Examples of problems in P . . . . .	236
7.3	The Class NP . . . . .	241

	Examples of problems in NP . . . . .	245
	The P versus NP question . . . . .	247
7.4	NP-completeness . . . . .	248
	Polynomial time reducibility . . . . .	249
	Definition of NP-completeness . . . . .	253
	The Cook–Levin Theorem . . . . .	254
7.5	Additional NP-complete Problems . . . . .	260
	The vertex cover problem . . . . .	261
	The Hamiltonian path problem . . . . .	262
	The subset sum problem . . . . .	268
	<i>Exercises and Problems</i> . . . . .	271
<b>8</b>	<b>Space Complexity</b> . . . . .	<b>277</b>
8.1	Savitch’s Theorem . . . . .	279
8.2	The Class PSPACE . . . . .	281
8.3	PSPACE-completeness . . . . .	283
	The TQBF problem . . . . .	283
	Winning strategies for games . . . . .	287
	Generalized geography . . . . .	289
8.4	The Classes L and NL . . . . .	294
8.5	NL-completeness . . . . .	297
	Searching in graphs . . . . .	298
8.6	NL equals coNL . . . . .	300
	<i>Exercises and Problems</i> . . . . .	302
<b>9</b>	<b>Intractability</b> . . . . .	<b>305</b>
9.1	Hierarchy Theorems . . . . .	306
	Exponential space completeness . . . . .	313
9.2	Relativization . . . . .	318
	Limits of the diagonalization method . . . . .	319
9.3	Circuit Complexity . . . . .	321
	<i>Exercises and Problems</i> . . . . .	330
<b>10</b>	<b>Advanced topics in complexity theory</b> . . . . .	<b>333</b>
10.1	Approximation Algorithms . . . . .	333
10.2	Probabilistic Algorithms . . . . .	335
	The class BPP . . . . .	336
	Primality . . . . .	339
	Read-once branching programs . . . . .	343
10.3	Alternation . . . . .	348
	Alternating time and space . . . . .	349
	The Polynomial time hierarchy . . . . .	353
10.4	Interactive Proof Systems . . . . .	354
	Graph nonisomorphism . . . . .	355
	Definition of the model . . . . .	355
	IP = PSPACE . . . . .	357

10.5	Parallel Computation . . . . .	366
	Uniform Boolean circuits . . . . .	367
	The class NC . . . . .	369
	P-completeness . . . . .	371
10.6	Cryptography . . . . .	372
	Secret keys . . . . .	372
	Public-key cryptosystems . . . . .	374
	One-way functions . . . . .	374
	Trapdoor functions . . . . .	376
	<i>Exercises and Problems</i> . . . . .	378
	<b>Selected Bibliography</b>	<b>381</b>
	<b>Index</b>	<b>387</b>