

Contents

Preface	xi
1 Introduction to the Theory of Computation	1
1.1 Mathematical Preliminaries and Notation	3
Sets	3
Functions and Relations	6
Graphs and Trees	8
Proof Techniques	10
1.2 Three Basic Concepts	16
Languages	16
Grammars	20
Automata	26
1.3 Some Applications*	30
2 Finite Automata	37
2.1 Deterministic Finite Accepters	38
Deterministic Accepters and Transition Graphs	38
Languages and Dfa's	40
Regular Languages	44
2.2 Nondeterministic Finite Accepters	49
Definition of a Nondeterministic Accepter	49
Why Nondeterminism?	53
2.3 Equivalence of Deterministic and Nondeterministic Finite Accepters	56
2.4 Reduction of the Number of States in Finite Automata*	63

3	Regular Languages and Regular Grammars	71
3.1	Regular Expressions	71
	Formal Definition of a Regular Expression	72
	Languages Associated with Regular Expressions	72
3.2	Connection Between Regular Expressions and Regular Languages	77
	Regular Expressions Denote Regular Languages	78
	Regular Expressions for Regular Languages	80
	Regular Expressions for Describing Simple Patterns	86
3.3	Regular Grammars	89
	Right- and Left-Linear Grammars	89
	Right-Linear Grammars Generate Regular Languages	91
	Right-Linear Grammars for Regular Languages	93
	Equivalence of Regular Languages and Regular Grammars	95
4	Properties of Regular Languages	99
4.1	Closure Properties of Regular Languages	100
	Closure under Simple Set Operations	100
	Closure under Other Operations	102
4.2	Elementary Questions about Regular Languages	111
4.3	Identifying Nonregular Languages	114
	Using the Pigeonhole Principle	114
	A Pumping Lemma	115
5	Context-Free Languages	125
5.1	Context-Free Grammars	126
	Examples of Context-Free Languages	127
	Leftmost and Rightmost Derivations	129
	Derivation Trees	130
	Relation Between Sentential Forms and Derivation Trees	132
5.2	Parsing and Ambiguity	136
	Parsing and Membership	136
	Ambiguity in Grammars and Languages	140
5.3	Context-Free Grammars and Programming Languages	146
6	Simplification of Context-Free Grammars and Normal Forms	149
6.1	Methods for Transforming Grammars	150
	A Useful Substitution Rule	150
	Removing Useless Productions	152
	Removing λ -Productions	156
	Removing Unit-Productions	158

6.2	Two Important Normal Forms	164
	Chomsky Normal Form	164
	Greibach Normal Form	167
6.3	A Membership Algorithm for Context-Free Grammars*	171
7	Pushdown Automata	175
7.1	Nondeterministic Pushdown Automata	176
	Definition of a Pushdown Automaton	176
	The Language Accepted by a Pushdown Automaton	180
7.2	Pushdown Automata and Context-Free Languages	185
	Pushdown Automata for Context-Free Languages	185
	Context-Free Grammars for Pushdown Automata	190
7.3	Deterministic Pushdown Automata and Deterministic Context-Free Languages	196
7.4	Grammars for Deterministic Context-Free Languages*	201
8	Properties of Context-Free Languages	205
8.1	Two Pumping Lemmas	205
	A Pumping Lemma for Context-Free Languages	206
	A Pumping Lemma for Linear Languages	210
8.2	Closure Properties and Decision Algorithms for Context- Free Languages	214
	Closure of Context-Free Languages	214
	Some Decidable Properties of Context-Free Languages	218
9	Turing Machines	223
9.1	The Standard Turing Machine	224
	Definition of a Turing Machine	224
	Turing Machines as Language Acceptors	231
	Turing Machines as Transducers	234
9.2	Combining Turing Machines for Complicated Tasks	240
9.3	Turing's Thesis	245
10	Other Models of Turing Machines	251
10.1	Minor Variations on the Turing Machine Theme	252
	Equivalence of Classes of Automata	252
	Turing Machines with a Stay-Option	253
	Turing Machines with Semi-Infinite Tape	255
	The Off-Line Turing Machine	257
10.2	Turing Machines with More Complex Storage	260
	Multitape Turing Machines	260
	Multidimensional Turing Machines	262
10.3	Nondeterministic Turing Machines	264
10.4	A Universal Turing Machine	268
10.5	Linear Bounded Automata	273

11	A Hierarchy of Formal Languages and Automata	277
11.1	Recursive and Recursively Enumerable Languages	278
	Languages That Are Not Recursively Enumerable	280
	A Language That Is Not Recursively Enumerable	281
	A Language That Is Recursively Enumerable but Not Recursive	283
11.2	Unrestricted Grammars	284
11.3	Context-Sensitive Grammars and Languages	291
	Context-Sensitive Languages and Linear Bounded Automata	292
	Relation Between Recursive and Context-Sensitive Languages	294
11.4	The Chomsky Hierarchy	296
12	Limits of Algorithmic Computation	299
12.1	Some Problems That Cannot Be Solved by Turing Machines	300
	Computability and Decidability	300
	The Turing Machine Halting Problem	301
	Reducing One Undecidable Problem to Another	304
12.2	Undecidable Problems for Recursively Enumerable Languages	308
12.3	The Post Correspondence Problem	311
12.4	Undecidable Problems for Context-Free Languages	318
12.5	A Question of Efficiency	322
13	Other Models of Computation	325
13.1	Recursive Functions	327
	Primitive Recursive Functions	328
	Ackermann's Function	331
	μ Recursive Functions	333
13.2	Post Systems	335
13.3	Rewriting Systems	339
	Matrix Grammars	340
	Markov Algorithms	340
	L-Systems	342
14	An Overview of Computational Complexity	345
14.1	Efficiency of Computation	346
14.2	Turing Machine Models and Complexity	348
14.3	Language Families and Complexity Classes	351
14.4	The Complexity Classes P and NP	355
14.5	Some NP Problems	356
14.6	Polynomial-Time Reduction	360
14.7	NP-Completeness and an Open Question	362

Appendix A	Finite-State Transducers	365
A.1	A General Framework	365
A.2	Mealy Machines	366
A.3	Moore Machines	368
A.4	Moore and Mealy Machine Equivalence	370
A.5	Mealy Machine Minimization	374
A.6	Moore Machine Minimization	378
A.7	Limitations of Finite-State Transducers	380
Appendix B	JFLAP: A Recommendation	383
Answers	Solutions and Hints for Selected Exercises	385
	References for Further Reading	431
	Index	433

