

Contents

Preface	xi
List of Acronyms	xv
I PROLEGOMENA	1
1 Introduction	3
1.1 The Three Pillars of Computation Theory	3
1.1.1 State	3
1.1.2 Encoding	4
1.1.3 Nondeterminism	7
1.2 The Nature of Computation Theory	10
2 Mathematical Preliminaries	13
2.1 Sets and Their Operations	13
2.2 Binary Relations	17
2.2.1 The Formal Notion of Binary Relation	17
2.2.2 Equivalence Relations	18
2.3 Functions	20
2.4 Formal Languages	22
2.4.1 The Notion of Language in Computation Theory	22
2.4.2 Languages as Metaphors for Computational Problems	24
2.5 Graphs and Trees	25
2.6 Useful Quantitative Notions	26
II STATE	31
3 Online Automata: Exemplars of “State”	33
3.1 Online Automata and Their “Languages”	33
3.2 A Myhill–Nerode-like Theorem for OAs	41
3.3 A Concrete OA: The Online Turing Machine	43

4	Finite Automata and Regular Languages	51
4.1	Introduction	51
4.2	Preliminaries	53
4.3	The Myhill–Nerode Theorem for FAs	56
4.3.1	The Theorem: States Are Equivalence Classes	57
4.3.2	What Do Equivalence Classes Look Like?	59
5	Applications of the Myhill–Nerode Theorem	63
5.1	Proving that Languages Are <i>Not</i> Regular	64
5.2	On Minimizing Finite Automata	66
5.3	Finite Automata with Probabilistic Transitions	70
5.3.1	PFAs and Their Languages	70
5.3.2	PFA Languages and Regular Languages	72
5.4	State as a Memory-Constraining Resource	79
5.4.1	$\mathcal{A}_M(n)$ for Two Specific Infinite OAs	80
5.4.2	A Bound on $\mathcal{A}_M(n)$ for Any OA M with Nonregular $L(M)$	81
5.5	State as a Time-Constraining Resource	83
5.5.1	Online TMs with Multiple Complex Tapes	84
5.5.2	An Information-Retrieval Problem as a Language	86
5.5.3	The Impact of Tape Structure on Memory Locality	87
5.5.4	Tape Dimensionality and the Time-Complexity of L_{DB}	88
6	Enrichment Topics	91
6.1	Pumping in Formal Languages	91
6.1.1	The Phenomenon of Pumping in Finite, Closed Systems	91
6.1.2	Pumping in Regular Languages	93
6.1.3	Pumping in Nonregular Languages	96
6.2	Closure Properties of the Regular Languages	101
6.3	Systems of Linear Equations with Languages as Coefficients	105
III	ENCODING	111
7	Countability and Uncountability: The Precursors of “Encoding”	113
7.1	Encoding Functions and Proofs of Countability	116
7.2	Diagonalization: Proofs of Uncountability	121
7.3	Where Has (Un)countability Led Us?	123
8	Enrichment Topic: “Efficient” Pairing Functions, with Applications	125
8.1	Background	126
8.2	The Prettiest Pairing Function(s)	127
8.2.1	The Diagonal PF $\mathcal{D}(x,y)$	127
8.2.2	Is $\mathcal{D}(x,y)$ the <i>Only</i> Polynomial PF?	129
8.3	Pairing Functions and the Storage of Extendible Arrays/Tables	130
8.3.1	Array-Storage Mappings via Pairing Functions	131
8.3.2	Pursuing <i>Compact</i> Pairing Functions	133
8.4	Pairing Functions and Volunteer Computing	139

8.4.1	A Methodology for Designing <i>Additive</i> Pairing Functions . . .	141
8.4.2	A Sampler of Explicit APFs	142
9	Computability Theory	147
9.1	Introduction and History	147
9.2	Preliminaries	151
9.2.1	Representing Computational Problems as Formal Languages .	151
9.2.2	Functions and Partial Functions	153
9.2.3	Self-Referential Programs: Interpreters and Compilers	155
9.3	The Halting Problem: The “Oldest” Unsolvable Problem	155
9.3.1	The Halting Problem Is Semisolvable but Not Solvable	156
9.3.2	Why We Care about the Halting Problem—An Example	158
9.4	Mapping Reducibility	160
9.4.1	Basic Properties of m -Reducibility	162
9.4.2	The s - m - n Theorem: Where Does One Find Encodings?	163
9.5	The Rice–Myhill–Shapiro Theorem	165
9.6	Complete, or “Hardest,” Semidecidable Problems	169
9.7	Some Important Limitations of Computability	172
9.8	(Online) Turing Machines and the Church–Turing Thesis	174
9.8.1	Simplifying an OTM without Diminishing Its Power	176
9.8.2	Augmented TMs That Are No More Powerful Than OTMs	195
IV	NONDETERMINISM	209
10	Nondeterministic Online Automata	211
10.1	Nondeterministic OAs	211
10.2	Nondeterminism as Unbounded Search, 1	213
10.3	An Overview of Nondeterminism in Computation Theory	215
11	Nondeterministic FAs	217
11.1	Nondeterministic FAs vs. Deterministic FAs	217
11.1.1	NFAs Are No More Powerful Than DFAs	217
11.1.2	Does the Subset Construction Waste DFA States?	219
11.2	An Application: The Kleene–Myhill Theorem	221
11.2.1	A Convenient Enhancement of NFAs	221
11.2.2	The Kleene–Myhill Theorem	223
12	Nondeterminism in Computability Theory	233
12.1	Introduction	233
12.2	Nondeterministic Turing Machines	234
12.2.1	The NTM Model	234
12.2.2	Deterministic Simulation of Nondeterminism: NTMs and OTMs	236
12.3	Nondeterminism as Unbounded Search, 2	241

13 Complexity Theory	245
13.1 Introduction	245
13.2 Time and Space Complexity	252
13.2.1 On Measuring Time Complexity	253
13.2.2 On Measuring Space Complexity	256
13.3 Reducibility, Hardness, and Completeness in Complexity Theory ...	263
13.3.1 A General Look at Resource-Bounded Computation	263
13.3.2 <i>Efficient</i> Mapping Reducibility	264
13.3.3 Hard Problems and Complete Problems	268
13.3.4 An NP -Complete Version of the Halting Problem	269
13.3.5 The Cook-Levin Theorem: The NP -Completeness of SAT ...	276
13.4 Nondeterminism and Space Complexity	285
13.4.1 Simulating Nondeterminism Space-Efficiently: Savitch's Theorem	287
13.4.2 Beyond Savitch's Theorem	297
V Sample Exercises	299
References	311
Index	315



<http://www.springer.com/978-0-387-09638-4>

The Pillars of Computation Theory
State, Encoding, Nondeterminism
Rosenberg, A.L.
2010, XVIII, 326 p. 49 illus., Softcover
ISBN: 978-0-387-09638-4