

Models of Computation
Exploring the Power of Computing

John E. Savage
Brown University

Contents

Preface vii

I Overview of the Book I

I The Role of Theory in Computer Science 3

1.1 A Brief History of Theoretical Computer Science 4

- 1.1.1 Early Years 4
- 1.1.2 1950s 5
- 1.1.3 1960s 5
- 1.1.4 1970s 5
- 1.1.5 1980s and 1990s 6

1.2 Mathematical Preliminaries 7

- 1.2.1 Sets 7
- 1.2.2 Number Systems 8
- 1.2.3 Languages and Strings 9
- 1.2.4 Relations 9
- 1.2.5 Graphs 10
- 1.2.6 Matrices 11
- 1.2.7 Functions 11
- 1.2.8 Rate of Growth of Functions 13

1.3 Methods of Proof 14

1.4	Computational Models	16
1.4.1	Logic Circuits	16
1.4.2	Finite-State Machines	18
1.4.3	Random-Access Machine	19
1.4.4	Other Models	20
1.4.5	Formal Languages	21
1.5	Computational Complexity	23
1.5.1	A Computational Inequality	23
1.5.2	Tradeoffs in Space, Time, and I/O Operations	24
1.5.3	Complexity Classes	26
1.5.4	Circuit Complexity	27
1.6	Parallel Computation	27
	Problems	29
	Chapter Notes	32

II General Computational Models 33

2	Logic Circuits	35
2.1	Designing Circuits	36
2.2	Straight-Line Programs and Circuits	36
2.2.1	Functions Computed by Circuits	38
2.2.2	Circuits That Compute Functions	39
2.2.3	Circuit Complexity Measures	40
2.2.4	Algebraic Properties of Boolean Functions	40
2.3	Normal-Form Expansions of Boolean Functions	42
2.3.1	Disjunctive Normal Form	42
2.3.2	Conjunctive Normal Form	43
2.3.3	SOPE and POSE Normal Forms	44
2.3.4	Ring-Sum Expansion	45
2.3.5	Comparison of Normal Forms	45
2.4	Reductions Between Functions	46
2.5	Specialized Circuits	47
2.5.1	Logical Operations	48
2.5.2	Shifting Functions	48
2.5.3	Encoder	51
2.5.4	Decoder	53
2.5.5	Multiplexer	54
2.5.6	Demultiplexer	55
2.6	Prefix Computations	55
2.6.1	An Efficient Parallel Prefix Circuit	57

2.7	Addition	58
2.7.1	Carry-Lookahead Addition	60
2.8	Subtraction	61
2.9	Multiplication	62
2.9.1	Carry-Save Multiplication	64
2.9.2	Divide-and-Conquer Multiplication	66
2.9.3	Fast Multiplication	67
2.9.4	Very Fast Multiplication	67
2.9.5	Reductions to Multiplication	68
2.10	Reciprocal and Division	68
2.10.1	Reductions to the Reciprocal	72
2.11	Symmetric Functions	74
2.12	Most Boolean Functions Are Complex	77
2.13	Upper Bounds on Circuit Size	79
	Problems	82
	Chapter Notes	88
3	Machines with Memory	91
3.1	Finite-State Machines	92
3.1.1	Functions Computed by FSMs	94
3.1.2	Computational Inequalities for the FSM	95
3.1.3	Circuits Are Universal for Bounded FSM Computations	96
3.1.4	Interconnections of Finite-State Machines	97
3.1.5	Nondeterministic Finite-State Machines	98
3.2	Simulating FSMs with Shallow Circuits*	100
3.2.1	A Shallow Circuit Simulating Addition	105
3.3	Designing Sequential Circuits	106
3.3.1	Binary Memory Devices	109
3.4	Random-Access Machines	110
3.4.1	The RAM Architecture	110
3.4.2	The Bounded-Memory RAM as FSM	111
3.4.3	Unbounded-Memory RAM Programs	112
3.4.4	Universality of the Unbounded-Memory RAM	114
3.5	Random-Access Memory Design	115
3.6	Computational Inequalities for the RAM	117
3.7	Turing Machines	118
3.7.1	Nondeterministic Turing Machines	120
3.8	Universality of the Turing Machine	121

3.9	Turing Machine Circuit Simulations	124
3.9.1	A Simple Circuit Simulation of TM Computations	124
3.9.2	Computational Inequalities for Turing Machines	127
3.9.3	Reductions from Turing to Circuit Computations	128
3.9.4	Definitions of P -Complete and NP -Complete Languages	130
3.9.5	Reductions to P -Complete Languages	130
3.9.6	Reductions to NP -Complete Languages	132
3.9.7	An Efficient Circuit Simulation of TM Computations*	134
3.10	Design of a Simple CPU	137
3.10.1	The Register Set	138
3.10.2	The Fetch-and-Execute Cycle	139
3.10.3	The Instruction Set	139
3.10.4	Assembly-Language Programming	140
3.10.5	Timing and Control	142
3.10.6	CPU Circuit Size and Depth	146
3.10.7	Emulation	147
	Problems	147
	Chapter Notes	152
4	Finite-State Machines and Pushdown Automata	153
4.1	Finite-State Machine Models	154
4.2	Equivalence of DFMSs and NFSMs	156
4.3	Regular Expressions	158
4.4	Regular Expressions and FSMs	160
4.4.1	Recognition of Regular Expressions by FSMs	160
4.4.2	Regular Expressions Describing FSM Languages	164
4.4.3	grep—Searching for Strings in Files	168
4.5	The Pumping Lemma for FSMs	168
4.6	Properties of Regular Languages	170
4.7	State Minimization*	171
4.7.1	Equivalence Relations on Languages and States	171
4.7.2	The Myhill-Nerode Theorem	174
4.7.3	A State Minimization Algorithm	175
4.8	Pushdown Automata	177
4.9	Formal Languages	181
4.9.1	Phrase-Structure Languages	182
4.9.2	Context-Sensitive Languages	183
4.9.3	Context-Free Languages	183
4.9.4	Regular Languages	184
4.10	Regular Language Recognition	184

4.11	Parsing Context-Free Languages	186
4.12	CFL Acceptance with Pushdown Automata*	192
4.13	Properties of Context-Free Languages	197
4.13.1	CFL Pumping Lemma	197
4.13.2	CFL Closure Properties	198
	Problems	200
	Chapter Notes	207
5	Computability	209
5.1	The Standard Turing Machine Model	210
5.1.1	Programming the Turing Machine	211
5.2	Extensions to the Standard Turing Machine Model	213
5.2.1	Multi-Tape Turing Machines	213
5.2.2	Nondeterministic Turing Machines	214
5.2.3	Oracle Turing Machines	216
5.2.4	Representing Restricted Models of Computation	217
5.3	Configuration Graphs	218
5.4	Phrase-Structure Languages and Turing Machines	219
5.5	Universal Turing Machines	220
5.6	Encodings of Strings and Turing Machines	222
5.7	Limits on Language Acceptance	223
5.7.1	Decidable Languages	223
5.7.2	A Language That Is Not Recursively Enumerable	224
5.7.3	Recursively Enumerable but Not Decidable Languages	225
5.8	Reducibility and Unsolvability	226
5.8.1	Reducibility	226
5.8.2	Unsolvable Problems	227
5.9	Functions Computed by Turing Machines	230
5.9.1	Primitive Recursive Functions	231
5.9.2	Partial Recursive Functions	232
5.9.3	Partial Recursive Functions are RAM-Computable	233
	Problems	233
	Chapter Notes	236
6	Algebraic and Combinatorial Circuits	237
6.1	Straight-Line Programs	238
6.2	Mathematical Preliminaries	239

6.2.1	Rings and Fields	239
6.2.2	Matrices	240
6.3	Matrix Multiplication	244
6.3.1	Strassen's Algorithm	245
6.4	Transitive Closure	248
6.5	Matrix Inversion	252
6.5.1	Symmetric Positive Definite Matrices	253
6.5.2	Schur Factorization	254
6.5.3	Inversion of Triangular Matrices	255
6.5.4	LDL ^T Factorization of SPD Matrices	257
6.5.5	Fast Matrix Inversion*	260
6.6	Solving Linear Systems	262
6.7	Convolution and the FFT Algorithm	263
6.7.1	Commutative Rings*	264
6.7.2	The Discrete Fourier Transform	264
6.7.3	Fast Fourier Transform	266
6.7.4	Convolution Theorem	268
6.8	Merging and Sorting Networks	270
6.8.1	Sorting Via Bitonic Merging	271
6.8.2	Fast Sorting Networks	274
	Problems	274
	Chapter Notes	278
7	Parallel Computation	281
7.1	Parallel Computational Models	282
7.2	Memoryless Parallel Computers	282
7.3	Parallel Computers with Memory	283
7.3.1	Flynn's Taxonomy	285
7.3.2	The Data-Parallel Model	286
7.3.3	Networked Computers	287
7.4	The Performance of Parallel Algorithms	289
7.4.1	Amdahl's Law	290
7.4.2	Brent's Principle	291
7.5	Multidimensional Meshes	292
7.5.1	Matrix-Vector Multiplication on a Linear Array	293
7.5.2	Sorting on Linear Arrays	294
7.5.3	Matrix Multiplication on a 2D Mesh	295
7.5.4	Embedding of 1D Arrays in 2D Meshes	297
7.6	Hypercube-Based Machines	298

7.6.1	Embedding Arrays in Hypercubes	299
7.6.2	Cube-Connected Cycles	300
7.7	Normal Algorithms	301
7.7.1	Summing on the Hypercube	302
7.7.2	Broadcasting on the Hypercube	303
7.7.3	Shifting on the Hypercube	303
7.7.4	Shuffle and Unshuffle Permutations on Linear Arrays	304
7.7.5	Fully Normal Algorithms on Two-Dimensional Arrays	306
7.7.6	Normal Algorithms on Cube-Connected Cycles	307
7.7.7	Fast Matrix Multiplication on the Hypercube	308
7.8	Routing in Networks	309
7.8.1	Local Routing Networks	309
7.8.2	Global Routing Networks	310
7.9	The PRAM Model	311
7.9.1	Simulating Trees, Arrays, and Hypercubes on the PRAM	313
7.9.2	The Power of Concurrency	314
7.9.3	Simulating the PRAM on a Hypercube Network	315
7.9.4	Circuits and the CREW PRAM	317
7.10	The BSP and LogP Models	317
	Problems	318
	Chapter Notes	322

III Computational Complexity 325

8	Complexity Classes	327
8.1	Introduction	328
8.2	Languages and Problems	328
8.2.1	Complements of Languages and Decision Problems	329
8.3	Resource Bounds	330
8.4	Serial Computational Models	331
8.4.1	The Random-Access Machine	331
8.4.2	Turing Machine Models	332
8.5	Classification of Decision Problems	334
8.5.1	Space and Time Hierarchies	336
8.5.2	Time-Bounded Complexity Classes	337
8.5.3	Space-Bounded Complexity Classes	338
8.5.4	Relations Between Time- and Space-Bounded Classes	341
8.5.5	Space-Bounded Functions	342
8.6	Complements of Complexity Classes	343

8.6.1	The Complement of NP	347
8.7	Reductions	349
8.8	Hard and Complete Problems	350
8.9	P-Complete Problems	352
8.10	NP-Complete Problems	355
8.10.1	NP-Complete Satisfiability Problems	356
8.10.2	Other NP-Complete Problems	357
8.11	The Boundary Between P and NP	363
8.12	PSPACE-Complete Problems	365
8.12.1	A First PSPACE-Complete Problem	365
8.12.2	Other PSPACE-Complete Problems	369
8.13	The Circuit Model of Computation	372
8.13.1	Uniform Families of Circuits	373
8.13.2	Uniform Circuits Are Equivalent to Turing Machines	374
8.14	The Parallel Random-Access Machine Model	376
8.14.1	Equivalence of the CREW PRAM and Circuits	376
8.14.2	The Parallel Computation Thesis	379
8.15	Circuit Complexity Classes	380
8.15.1	Efficiently Parallelizable Languages	380
8.15.2	Circuits of Polynomial Size	382
	Problems	383
	Chapter Notes	388
9	Circuit Complexity	391
9.1	Circuit Models and Measures	392
9.1.1	Circuit Models	392
9.1.2	Complexity Measures	393
9.2	Relationships Among Complexity Measures	394
9.2.1	Effect of Fan-Out on Circuit Size	394
9.2.2	Effect of Basis Change on Circuit Size and Depth	396
9.2.3	Formula Size Versus Circuit Depth	396
9.3	Lower-Bound Methods for General Circuits	399
9.3.1	Simple Lower Bounds	399
9.3.2	The Gate-Elimination Method for Circuit Size	400
9.4	Lower-Bound Methods for Formula Size	404
9.4.1	The Nečiporuk Lower Bound	405
9.4.2	The Krapchenko Lower Bound	407
9.5	The Power of Negation	409

9.6	Lower-Bound Methods for Monotone Circuits	412
9.6.1	The Path-Elimination Method	413
9.6.2	The Function Replacement Method	417
9.6.3	The Approximation Method	424
9.6.4	Slice Functions	431
9.7	Circuit Depth	436
9.7.1	Communication Complexity	437
9.7.2	General Depth and Communication Complexity	438
9.7.3	Monotone Depth and Communication Complexity	440
9.7.4	The Monotone Depth of the Clique Function	442
9.7.5	Bounded-Depth Circuits	447
	Problems	450
	Chapter Notes	455
10	Space—Time Tradeoffs	461
10.1	The Pebble Game	462
10.1.1	The Pebble Game Versus the Branching Program	462
10.1.2	Playing the Pebble Game	463
10.2	Space Lower Bounds	464
10.3	Extreme Tradeoffs	466
10.4	Grigoriev’s Lower-Bound Method	468
10.4.1	Flow Properties of Functions	468
10.4.2	The Lower-Bound Method in the Basic Pebble Game	470
10.4.3	First Matrix Multiplication Bound	472
10.5	Applications of Grigoriev’s Method	472
10.5.1	Convolution	473
10.5.2	Cyclic Shifting	474
10.5.3	Integer Multiplication	475
10.5.4	Matrix Multiplication	476
10.5.5	Discrete Fourier Transform	479
10.5.6	Merging Networks	481
10.6	Worst-Case Tradeoffs for Pebble Games*	482
10.7	Upper Bounds on Space*	483
10.8	Lower Bound on Space for General Graphs*	484
10.9	Branching Programs	488
10.9.1	Branching Programs and Other Models	493
10.10	Straight-Line Versus Branching Programs	495
10.10.1	Efficient Branching Programs for Cyclic Shift	496
10.10.2	Efficient Branching Programs for Merging	496

10.11	The Borodin-Cook Lower-Bound Method	497
10.12	Properties of “nice” and “ok” Matrices*	501
10.13	Applications of the Borodin-Cook Method	504
10.13.1	Convolution	505
10.13.2	Integer Multiplication	506
10.13.3	Matrix-Vector Product	507
10.13.4	Matrix Multiplication*	509
10.13.5	Matrix Inversion	511
10.13.6	Discrete Fourier Transform	513
10.13.7	Unique Elements	514
10.13.8	Sorting	517
	Problems	519
	Chapter Notes	526
11	Memory-Hierarchy Tradeoffs	529
11.1	The Red-Blue Pebble Game	530
11.1.1	Playing the Red-Blue Pebble Game	532
11.1.2	Balanced Computer Systems	532
11.2	The Memory-Hierarchy Pebble Game	533
11.2.1	Playing the MHG	535
11.3	I/O-Time Relationships	535
11.4	The Hong-Kung Lower-Bound Method	537
11.5	Tradeoffs Between Space and I/O Time	539
11.5.1	Matrix-Vector Product	539
11.5.2	Matrix-Matrix Multiplication	541
11.5.3	The Fast Fourier Transform	546
11.5.4	Convolution	552
11.6	Block I/O in the MHG	555
11.7	Simulating a Fast Memory in the MHG	558
11.8	RAM-Based I/O Models	559
11.8.1	The Block-Transfer Model	559
11.9	The Hierarchical Memory Model	563
11.9.1	Lower Bounds for the HMM	564
11.9.2	Upper Bounds for the HMM	567
11.10	Competitive Memory Management	567
11.10.1	Two-Level Memory-Management Algorithms	568
	Problems	569
	Chapter Notes	573

12	VLSI Models of Computation	575
12.1	The VLSI Challenge	576
12.1.1	Chip Fabrication	576
12.1.2	Design and Layout	577
12.2	VLSI Physical Models	578
12.3	VLSI Computational Models	579
12.4	VLSI Performance Criteria	580
12.5	Chip Layout	581
12.5.1	The H-Tree Layout	581
12.5.2	Multi-dimensional Mesh Layouts	583
12.5.3	Layout of the CCC Network	584
12.6	Area-Time Tradeoffs	586
12.6.1	Planar Circuit Size	586
12.6.2	Computational Inequalities	587
12.6.3	The Planar Separator Theorem	589
12.7	The Performance of VLSI Algorithms	592
12.7.1	The Performance of VLSI Algorithms on Functions	593
12.7.2	The Performance of VLSI Algorithms on Predicates	595
12.8	Area Bounds	597
	Problems	598
	Chapter Notes	601
	Bibliography	605
Index	623	