

Contents

1	Limits? What Limits?	1
1.1	Physical Limits of Computation	2
1.1.1	Fundamental Engineering Constraints to Semiconductor Manufacturing and Scaling	2
1.1.2	Fundamental Limits to Energy Efficiency	3
1.1.3	Fundamental Physical Constraints on Computing in General	3
1.2	The Limits Addressed	4
1.2.1	Computability Overview	4
1.2.2	Complexity Overview	6
	References	9
 Part I Computability		
2	Problems and Effective Procedures	13
2.1	On Computability	14
2.1.1	Historical Remarks	14
2.1.2	Effective Procedures	16
2.2	Sets, Relations and Functions	17
2.2.1	Sets	17
2.2.2	Relations	21
2.2.3	Functions	22
2.2.4	Partial Functions	22
2.2.5	Total Functions	23
2.3	Problems	24
2.3.1	Computing Solutions to Problems	26
	References	27

3	The <code>WHILE</code>-Language	29
3.1	The Data Type of Binary Trees	31
3.2	<code>WHILE</code> -Syntax	32
3.2.1	Expressions	32
3.2.2	Commands	32
3.2.3	Programs	33
3.2.4	A Grammar for <code>WHILE</code>	33
3.2.5	Layout Conventions and Brackets	34
3.3	Encoding Data Types as Trees	35
3.3.1	Boolean Values	35
3.3.2	Lists and Pairs	36
3.3.3	Natural Numbers	37
3.3.4	Finite Words	40
3.4	Sample Programs	40
3.4.1	Addition	40
3.4.2	List Reversal	41
3.4.3	Tail Recursion	42
3.4.4	Analysis of Algorithms	43
	References	45
4	Semantics of <code>WHILE</code>	47
4.1	Stores	48
4.2	Semantics of Programs	49
4.3	Semantics of Commands	50
4.4	Semantics of Expressions	52
	References	54
5	Extensions of <code>WHILE</code>	55
5.1	Equality	55
5.2	Literals	56
5.2.1	Number Literals	56
5.2.2	Boolean Literals	57
5.3	Adding Atoms	57
5.4	List Constructor	58
5.5	Macro Calls	59
5.6	Switch Statement	60
	References	63
6	Programs as Data Objects	65
6.1	Interpreters Formally	66
6.2	Abstract Syntax Trees	67
6.3	Encoding of <code>WHILE</code> -ASTs in \mathbb{D}	67
	Reference	70

7	A Self-interpreter for WHILE	71
7.1	A Self-interpreter for WHILE -Programs with One Variable	72
7.1.1	General Tree Traversal for ASTs	72
7.1.2	The STEP Macro	73
7.2	A Self-interpreter for WHILE	81
7.2.1	Store Manipulation Macros	83
	References	86
8	An Undecidable (Non-computable) Problem	87
8.1	WHILE-Computability and Decidability	87
8.2	The Halting Problem for WHILE	89
8.3	Diagonalisation and the Barber “Paradox”	90
8.4	Proof of the Undecidability of the Halting Problem	92
	References	95
9	More Undecidable Problems	97
9.1	Semi-decidability of the Halting Problem	97
9.2	Rice’s Theorem	99
9.3	The Tiling Problem	101
9.4	Problem Reduction	103
9.5	Other (Famous) Undecidable Problems	105
9.6	Dealing with Undecidable Problems	106
9.7	A Fast-Growing Non-computable Function	107
	References	111
10	Self-referencing Programs	113
10.1	The S-m-n Theorem	114
10.2	Kleene’s Recursion Theorem	116
10.3	Recursion Elimination	118
	References	121
11	The Church-Turing Thesis	123
11.1	The Thesis	124
11.2	Semantic Framework for Machine-Like Models	125
11.3	Turing Machines TM	126
11.4	GOTO-Language	129
11.5	Register Machines RAM and SRAM	131
11.6	Counter Machines CM	134
11.7	Cellular Automata	135
11.7.1	2D: Game of Life	138
11.7.2	1D: Rule 110	140
11.8	Robustness of Computability	141
11.8.1	The Crucial Role of Compilers	141
11.8.2	Equivalence of Models	142
	References	147

Part II Complexity

12 Measuring Time Usage.	151
12.1 Unit-Cost Time Measure	152
12.2 Time Measure for WHILE	154
12.3 Comparing Programming Languages Considering Time	157
References	160
13 Complexity Classes.	161
13.1 Runtime Bounds	162
13.2 Time Complexity Classes	163
13.3 Lifting Simulation Properties to Complexity Classes	165
13.4 Big-O and Little-o	166
References	171
14 Robustness of P	173
14.1 Extended Church–Turing Thesis	174
14.2 Invariance or Cook’s Thesis	174
14.2.1 Non-sequential Models	175
14.2.2 Evidence for Cook’s Thesis	176
14.2.3 Linear Time	178
14.3 Cobham–Edmonds Thesis	179
References	181
15 Hierarchy Theorems	183
15.1 Linear Time Hierarchy Theorems	184
15.2 Beyond Linear Time	189
15.3 Gaps in the Hierarchy	192
References	193
16 Famous Problems in P	195
16.1 Decision Versus Optimisation Problems	197
16.2 Predecessor Problem	198
16.3 Membership Test for a Context Free Language	201
16.4 Primality Test	202
16.5 Graph Problems	203
16.5.1 Reachability in a Graph	203
16.5.2 Shortest Paths in a Graph	204
16.5.3 Maximal Matchings	206
16.5.4 Min-Cut and Max-Flow	207
16.5.5 The Seven Bridges of Königsberg	208
16.6 Linear Programming	210
References	215
17 Common Problems Not Known to Be in P	217
17.1 The Travelling Salesman Problem (TSP)	218
17.2 The Graph Colouring Problem	220

17.3	Max-Cut Problem	221
17.4	The 0-1 Knapsack Problem	222
17.5	Integer Programming Problem	223
17.6	Does Not Being in P Matter?	224
	References	226
18	The One-Million-Dollar Question	227
18.1	The Complexity Class NP	228
18.2	Nondeterministic Programs	229
18.2.1	Time Measure of Nondeterministic Programs	231
18.2.2	Some Basic Facts About NP	233
18.3	Robustness of NP	234
18.4	Problems in NP	235
18.5	The Biggest Open Problem in (Theoretical) Computer Science	237
	References	239
19	How Hard Is a Problem?	241
19.1	Reminder: Effective Reductions	242
19.2	Polynomial Time Reduction	242
19.3	Hard Problems	245
	References	249
20	Complete Problems	251
20.1	A First NP -complete Problem	252
20.2	More NP -complete Problems	255
20.3	Puzzles and Games	256
20.3.1	Chess	258
20.3.2	Sudoku	259
20.3.3	Tile-Matching Games	260
20.4	Database Queries	261
20.5	Policy Based Routing	264
20.6	“Limbo” Problems	266
20.7	Complete Problems in Other Classes	268
20.7.1	P -complete	268
20.7.2	RE -complete	269
	References	273
21	How to Solve NP-Complete Problems	275
21.1	Exact Algorithms	276
21.2	Approximation Algorithms	276
21.3	Parallelism	281
21.4	Randomization	282
21.4.1	The Class RP	282
21.4.2	Probabilistic Algorithms	284

21.5	Solving the Travelling Salesman Problem	285
21.5.1	Exact Solutions	285
21.5.2	Approximative Solutions	286
21.6	When Bad Complexity is Good News.	289
	References	295
22	Molecular Computing.	299
22.1	The Beginnings of DNA Computing.	300
22.2	DNA Computing Potential.	301
22.3	DNA Computing Challenges	302
22.4	Abstract Models of Molecular Computation.	302
22.4.1	Chemical Reaction Networks (CRN)	303
22.4.2	CRNs as Effective Procedures.	305
22.4.3	Are CRNs Equivalent to Other Notions of Computation?	308
22.4.4	Time Complexity for CRNs	309
22.4.5	Implementing CRNs	309
	References	315
23	Quantum Computing	317
23.1	Molecular Electronics	318
23.2	The Mathematics of Quantum Mechanics	319
23.3	Quantum Computability and Complexity.	321
23.4	Quantum Algorithms	323
23.4.1	Shor's Algorithm	323
23.4.2	Grover's Algorithm	324
23.5	Building Quantum Computers	325
23.6	Quantum Computing Challenges	325
23.7	To Boldly Go	326
	References	328
	Further Reading—Computability and Complexity Textbooks	331
	Glossary	335
	Index	341

Limits of Computation

From a Programming Perspective

Reus, B.

2016, XVIII, 348 p. 80 illus., Softcover

ISBN: 978-3-319-27887-2