

Contents

Preface	3
1 Text Compression	9
1.1 Introduction	9
1.2 A Very Incomplete Introduction to Information Theory	10
1.3 Huffman Coding	13
1.3.1 Uniquely Decodable Codes and Prefix Codes	14
1.3.2 Constructing An Optimal Prefix Code	15
1.3.3 Proof of Correctness	17
1.3.4 Implementation	20
1.4 Lempel-Ziv Coding	20
2 Error Detection and Correction	25
2.1 Introduction	25
2.1.1 Properties of Exclusive-Or	26
2.1.2 Dependent Set	26
2.2 Small Applications	31
2.2.1 Complementation	31
2.2.2 Toggling	31
2.2.3 Exchange	32
2.2.4 Storage for Doubly-Linked Lists	32
2.2.5 The Game of Nim	33
2.3 Secure Communication	35
2.4 Oblivious Transfer	36
2.5 RAID Architecture	37
2.6 Error Detection	37
2.6.1 Parity Check Code	38
2.6.2 Horizontal and Vertical Parity Check	39
2.7 Error Correction	40
2.7.1 Hamming Distance	40
2.7.2 A Naive Error-Correcting Code	41
2.7.3 Hamming Code	43
2.7.4 Reed-Muller Code	46

3	Cryptography	51
3.1	Introduction	51
3.2	Early Encryption Schemes	52
3.2.1	Substitution Cyphers	52
3.2.2	Electronic Transmission	54
3.3	Public Key Cryptography	55
3.3.1	Mathematical Preliminaries	56
3.3.2	The RSA Scheme	60
3.4	Digital Signatures and Related Topics	65
3.5	Block Cipher	68
4	Finite State Machines	71
4.1	Introduction	71
4.1.1	Wolf-Goat-Cabbage Puzzle	71
4.1.2	A Traffic Light	74
4.1.3	A Pattern Matching Problem	74
4.2	Finite State Machine	76
4.2.1	What is it?	76
4.2.2	Reasoning about Finite State Machines	81
4.2.3	Finite State Transducers	83
4.2.4	Serial Binary Adder	85
4.2.5	Parity Generator	86
4.3	Specifying Control Logic Using Finite State Machines	88
4.3.1	The Game of Simon	88
4.3.2	Soda Machine	89
4.4	Regular Expressions	90
4.4.1	What is a Regular Expression?	91
4.4.2	Examples of Regular Expressions	92
4.4.3	Algebraic Properties of Regular Expressions	93
4.4.4	Solving Regular Expression Equations	95
4.4.5	From Regular Expressions to Machines	97
4.4.6	Regular Expressions in Practice; from GNU Emacs	99
4.5	Enhancements to Finite State Machines	102
4.5.1	Adding Structures to Transitions	103
4.5.2	Examples of Structured Transitions	104
4.5.3	Adding Structures to States	108
4.5.4	Examples of Structured States	111
5	Recursion and Induction	119
5.1	Introduction	119
5.2	Preliminaries	119
5.2.1	Running Haskell programs from command line	119
5.2.2	Loading Program Files	120
5.2.3	Comments	120
5.2.4	Program Layout	121
5.3	Primitive Data Types	121

5.3.1	Integer	122
5.3.2	Boolean	122
5.3.3	Character and String	123
5.4	Writing Function Definitions	124
5.4.1	Function Parameters and Binding	124
5.4.2	Examples of Function Definitions	125
5.4.3	Conditionals	127
5.4.4	The <code>where</code> Clause	128
5.4.5	Pattern Matching	128
5.5	Recursive Programming	129
5.5.1	Computing Powers of 2	130
5.5.2	Counting the 1s in a Binary Expansion	131
5.5.3	Multiplication via Addition	131
5.5.4	Fibonacci Numbers	132
5.5.5	Greatest Common Divisor	133
5.6	Tuple	134
5.7	Type	137
5.7.1	Polymorphism	138
5.7.2	Type Classes	139
5.7.3	Type Violation	140
5.8	List	140
5.8.1	The Type of a List	141
5.8.2	The List Constructor <i>Cons</i>	141
5.8.3	Pattern Matching on Lists	142
5.8.4	Recursive Programming on Lists	142
5.8.5	Mutual Recursion	145
5.9	Examples of Programming with Lists	147
5.9.1	Some Useful List Operations	147
5.9.2	Towers of Hanoi	150
5.9.3	Gray Code	151
5.9.4	Sorting	154
5.9.5	Polynomial Evaluation	161
5.10	Proving Facts about Recursive Programs	161
5.11	Higher Order Functions	163
5.11.1	Function <code>foldr</code>	163
5.11.2	Function <code>map</code>	165
5.11.3	Function <code>filter</code>	166
5.12	Program Design: Boolean Satisfiability	167
5.12.1	Boolean Satisfiability	167
5.12.2	Program Development	169
5.12.3	Variable Ordering	172
5.13	A Real-World Application: Google's Map-Reduce	174
5.13.1	Some Example Problems	175
5.13.2	Parallel Implementation and Empirical Results	176

6	Relational Database	179
6.1	Introduction	179
6.2	The Relational Data Model	181
6.2.1	Relations in Mathematics	181
6.2.2	Relations in Databases	182
6.3	Relational Algebra	183
6.3.1	Operations on Database Relations	183
6.3.2	Identities of Relational Algebra	187
6.3.3	Example of Query Optimization	188
6.3.4	Additional Operations on Relations	190
7	String Matching	193
7.1	Introduction	193
7.2	Rabin-Karp Algorithm	194
7.3	Knuth-Morris-Pratt Algorithm	196
7.3.1	Informal Description	197
7.3.2	Algorithm Outline	197
7.3.3	The Theory of Core	199
7.3.4	Computing Cores of all Non-empty Prefixes	202
7.4	Boyer-Moore Algorithm	204
7.4.1	Algorithm Outline	204
7.4.2	The Bad Symbol Heuristic	205
7.4.3	The Good Suffix Heuristic	207
8	Parallel Recursion	215
8.1	Parallelism and Recursion	215
8.2	Powerlist	215
8.2.1	Definitions	216
8.2.2	Functions over Powerlists	217
8.2.3	Discussion	219
8.3	Laws	219
8.4	Examples	221
8.4.1	Permutations	221
8.4.2	Reduction	224
8.4.3	Gray Code	224
8.4.4	Polynomial	225
8.4.5	Fast Fourier Transform	225
8.4.6	Batcher Sort	228
8.4.7	Prefix Sum	232
8.5	Higher Dimensional Arrays	237
8.5.1	Pointwise Application	240
8.5.2	Deconstruction	241
8.5.3	Embedding Arrays in Hypercubes	242
8.6	Remarks	243