
Contents

	<i>page ix</i>
<i>Preface</i>	
1 What is functional programming?	1
1.1 Functions and types	1
1.2 Functional composition	3
1.3 Example: common words	3
1.4 Example: numbers into words	7
1.5 The Haskell Platform	12
1.6 Exercises	14
1.7 Answers	17
1.8 Chapter notes	20
2 Expressions, types and values	22
2.1 A session with GHCi	22
2.2 Names and operators	25
2.3 Evaluation	27
2.4 Types and type classes	30
2.5 Printing values	33
2.6 Modules	35
2.7 Haskell layout	36
2.8 Exercises	37
2.9 Answers	42
2.10 Chapter notes	47
3 Numbers	49
3.1 The type class Num	49
3.2 Other numeric type classes	50
3.3 Computing floors	52
3.4 Natural numbers	56

vi	Contents	
	3.5 Exercises	59
	3.6 Answers	61
	3.7 Chapter notes	62
4	Lists	63
	4.1 List notation	63
	4.2 Enumerations	65
	4.3 List comprehensions	66
	4.4 Some basic operations	68
	4.5 Concatenation	69
	4.6 <code>concat</code> , <code>map</code> and <code>filter</code>	70
	4.7 <code>zip</code> and <code>zipWith</code>	73
	4.8 Common words, completed	75
	4.9 Exercises	77
	4.10 Answers	82
	4.11 Chapter notes	87
5	A simple Sudoku solver	89
	5.1 Specification	89
	5.2 Lawful program construction	95
	5.3 Pruning the matrix of choices	97
	5.4 Expanding a single cell	101
	5.5 Exercises	105
	5.6 Answers	107
	5.7 Chapter notes	109
6	Proofs	110
	6.1 Induction over natural numbers	110
	6.2 Induction over lists	113
	6.3 The function <code>foldr</code>	117
	6.4 The function <code>foldl</code>	122
	6.5 The function <code>scanl</code>	125
	6.6 The maximum segment sum	127
	6.7 Exercises	131
	6.8 Answers	135
	6.9 Chapter notes	144
7	Efficiency	145
	7.1 Lazy evaluation	145
	7.2 Controlling space	149
	7.3 Controlling time	154
	7.4 Analysing time	156
	7.5 Accumulating parameters	159

	Contents	vii
7.6	Tupling	164
7.7	Sorting	167
7.8	Exercises	172
7.9	Answers	175
7.10	Chapter notes	180
8	Pretty-printing	181
8.1	Setting the scene	181
8.2	Documents	183
8.3	A direct implementation	187
8.4	Examples	189
8.5	The best layout	191
8.6	A term representation	193
8.7	Exercises	199
8.8	Answers	203
8.9	Chapter notes	209
9	Infinite lists	210
9.1	Review	210
9.2	Cyclic lists	212
9.3	Infinite lists as limits	215
9.4	Paper–rock–scissors	221
9.5	Stream-based interaction	226
9.6	Doubly-linked lists	228
9.7	Exercises	231
9.8	Answers	234
9.9	Chapter notes	237
10	Imperative functional programming	239
10.1	The IO monad	239
10.2	More monads	244
10.3	The State monad	247
10.4	The ST monad	251
10.5	Mutable arrays	254
10.6	Immutable arrays	259
10.7	Exercises	263
10.8	Answers	267
10.9	Chapter notes	275
11	Parsing	276
11.1	Parsers as monads	276
11.2	Basic parsers	279
11.3	Choice and repetition	281

viii	Contents	
	11.4 Grammars and expressions	285
	11.5 Showing expressions	288
	11.6 Exercises	291
	11.7 Answers	294
	11.8 Chapter notes	297
12	A simple equational calculator	298
	12.1 Basic considerations	298
	12.2 Expressions	304
	12.3 Laws	310
	12.4 Calculations	312
	12.5 Rewrites	315
	12.6 Matchings	317
	12.7 Substitutions	319
	12.8 Testing the calculator	321
	12.9 Exercises	331
	12.10 Answers	333
	12.11 Chapter notes	337
	<i>Index</i>	338