Contents

Foreword xiii
Preface xv

Part I Basic Concepts 1

1 Introduction 3
1.1 Functions 3
1.2 Functional programming 4
1.3 Features of Haskell 6
1.4 Historical background 8
1.5 A taste of Haskell 9
1.6 Chapter remarks 13
1.7 Exercises 13

2 First steps 14
2.1 Glasgow Haskell Compiler 14
2.2 Installing and starting 14
2.3 Standard prelude 15
2.4 Function application 16
2.5 Haskell scripts 17
2.6 Chapter remarks 21
2.7 Exercises 21

3 Types and classes 22
3.1 Basic concepts 22
3.2 Basic types 23
3.3 List types 25
3.4 Tuple types 26
3.5 Function types 27
3.6 Curried functions 27
3.7 Polymorphic types 29
3.8 Overloaded types 30
3.9 Basic classes 31
3.10 Chapter remarks 36
Table of Contents

3.11 Exercises 36

4 Defining functions 38
4.1 New from old 38
4.2 Conditional expressions 38
4.3 Guarded equations 39
4.4 Pattern matching 40
4.5 Lambda expressions 42
4.6 Operator sections 44
4.7 Chapter remarks 45
4.8 Exercises 45

5 List comprehensions 47
5.1 Basic concepts 47
5.2 Guards 48
5.3 The zip function 50
5.4 String comprehensions 51
5.5 The Caesar cipher 52
5.6 Chapter remarks 56
5.7 Exercises 57

6 Recursive functions 59
6.1 Basic concepts 59
6.2 Recursion on lists 61
6.3 Multiple arguments 63
6.4 Multiple recursion 64
6.5 Mutual recursion 65
6.6 Advice on recursion 66
6.7 Chapter remarks 71
6.8 Exercises 71

7 Higher-order functions 73
7.1 Basic concepts 73
7.2 Processing lists 74
7.3 The foldr function 76
7.4 The foldl function 79
7.5 The composition operator 81
7.6 Binary string transmitter 82
7.7 Voting algorithms 86
7.8 Chapter remarks 89
7.9 Exercises 89

8 Declaring types and classes 92
8.1 Type declarations 92
8.2 Data declarations 93
8.3 Newtype declarations 95
8.4 Recursive types 96
8.5 Class and instance declarations 99
8.6 Tautology checker 101
8.7 Abstract machine 106
8.8 Chapter remarks 108
8.9 Exercises 109

9 The countdown problem 111
9.1 Introduction 111
9.2 Arithmetic operators 112
9.3 Numeric expressions 113
9.4 Combinatorial functions 114
9.5 Formalising the problem 115
9.6 Brute force solution 115
9.7 Performance testing 116
9.8 Combining generation and evaluation 117
9.9 Exploiting algebraic properties 118
9.10 Chapter remarks 119
9.11 Exercises 120

Part II Going Further 121

10 Interactive programming 123
10.1 The problem 123
10.2 The solution 124
10.3 Basic actions 125
10.4 Sequencing 126
10.5 Derived primitives 127
10.6 Hangman 128
10.7 Nim 129
10.8 Life 133
10.9 Chapter remarks 137
10.10 Exercises 137

11 Unbeatable tic-tac-toe 139
11.1 Introduction 139
11.2 Basic declarations 140
11.3 Grid utilities 141
11.4 Displaying a grid 142
11.5 Making a move 143
11.6 Reading a number 144
11.7 Human vs human 144
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.8</td>
<td>Game trees</td>
<td>145</td>
</tr>
<tr>
<td>11.9</td>
<td>Pruning the tree</td>
<td>147</td>
</tr>
<tr>
<td>11.10</td>
<td>Minimax algorithm</td>
<td>148</td>
</tr>
<tr>
<td>11.11</td>
<td>Human vs computer</td>
<td>150</td>
</tr>
<tr>
<td>11.12</td>
<td>Chapter remarks</td>
<td>151</td>
</tr>
<tr>
<td>11.13</td>
<td>Exercises</td>
<td>151</td>
</tr>
<tr>
<td>12</td>
<td>Monads and more</td>
<td>153</td>
</tr>
<tr>
<td>12.1</td>
<td>Functors</td>
<td>153</td>
</tr>
<tr>
<td>12.2</td>
<td>Applicatives</td>
<td>157</td>
</tr>
<tr>
<td>12.3</td>
<td>Monads</td>
<td>164</td>
</tr>
<tr>
<td>12.4</td>
<td>Chapter remarks</td>
<td>174</td>
</tr>
<tr>
<td>12.5</td>
<td>Exercises</td>
<td>175</td>
</tr>
<tr>
<td>13</td>
<td>Monadic parsing</td>
<td>177</td>
</tr>
<tr>
<td>13.1</td>
<td>What is a parser?</td>
<td>177</td>
</tr>
<tr>
<td>13.2</td>
<td>Parsers as functions</td>
<td>177</td>
</tr>
<tr>
<td>13.3</td>
<td>Basic definitions</td>
<td>179</td>
</tr>
<tr>
<td>13.4</td>
<td>Sequencing parsers</td>
<td>179</td>
</tr>
<tr>
<td>13.5</td>
<td>Making choices</td>
<td>181</td>
</tr>
<tr>
<td>13.6</td>
<td>Derived primitives</td>
<td>183</td>
</tr>
<tr>
<td>13.7</td>
<td>Handling spacing</td>
<td>186</td>
</tr>
<tr>
<td>13.8</td>
<td>Arithmetic expressions</td>
<td>187</td>
</tr>
<tr>
<td>13.9</td>
<td>Calculator</td>
<td>191</td>
</tr>
<tr>
<td>13.10</td>
<td>Chapter remarks</td>
<td>194</td>
</tr>
<tr>
<td>13.11</td>
<td>Exercises</td>
<td>194</td>
</tr>
<tr>
<td>14</td>
<td>Foldables and friends</td>
<td>196</td>
</tr>
<tr>
<td>14.1</td>
<td>Monoids</td>
<td>196</td>
</tr>
<tr>
<td>14.2</td>
<td>Foldables</td>
<td>200</td>
</tr>
<tr>
<td>14.3</td>
<td>Traversables</td>
<td>206</td>
</tr>
<tr>
<td>14.4</td>
<td>Chapter remarks</td>
<td>210</td>
</tr>
<tr>
<td>14.5</td>
<td>Exercises</td>
<td>210</td>
</tr>
<tr>
<td>15</td>
<td>Lazy evaluation</td>
<td>212</td>
</tr>
<tr>
<td>15.1</td>
<td>Introduction</td>
<td>212</td>
</tr>
<tr>
<td>15.2</td>
<td>Evaluation strategies</td>
<td>213</td>
</tr>
<tr>
<td>15.3</td>
<td>Termination</td>
<td>216</td>
</tr>
<tr>
<td>15.4</td>
<td>Number of reductions</td>
<td>217</td>
</tr>
<tr>
<td>15.5</td>
<td>Infinite structures</td>
<td>219</td>
</tr>
<tr>
<td>15.6</td>
<td>Modular programming</td>
<td>220</td>
</tr>
<tr>
<td>15.7</td>
<td>Strict application</td>
<td>223</td>
</tr>
<tr>
<td>15.8</td>
<td>Chapter remarks</td>
<td>226</td>
</tr>
<tr>
<td>15.9</td>
<td>Exercises</td>
<td>226</td>
</tr>
</tbody>
</table>
# Contents

16 **Reasoning about programs**

- 16.1 Equational reasoning 228
- 16.2 Reasoning about Haskell 229
- 16.3 Simple examples 230
- 16.4 Induction on numbers 231
- 16.5 Induction on lists 234
- 16.6 Making append vanish 238
- 16.7 Compiler correctness 241
- 16.8 Chapter remarks 246
- 16.9 Exercises 246

17 **Calculating compilers**

- 17.1 Introduction 249
- 17.2 Syntax and semantics 249
- 17.3 Adding a stack 250
- 17.4 Adding a continuation 252
- 17.5 Defunctionalising 254
- 17.6 Combining the steps 257
- 17.7 Chapter remarks 261
- 17.8 Exercises 261

Appendix A **Selected solutions**

- A.1 Introduction 263
- A.2 First steps 264
- A.3 Types and classes 265
- A.4 Defining functions 266
- A.5 List comprehensions 267
- A.6 Recursive functions 267
- A.7 Higher-order functions 268
- A.8 Declaring types and classes 269
- A.9 The countdown problem 270
- A.10 Interactive programming 270
- A.11 Unbeatable tic-tac-toe 271
- A.12 Monads and more 272
- A.13 Monadic parsing 273
- A.14 Foldables and friends 274
- A.15 Lazy evaluation 275
- A.16 Reasoning about programs 276
- A.17 Calculating compilers 279

Appendix B **Standard prelude**

- B.1 Basic classes 280
- B.2 Booleans 281
- B.3 Characters 282
Contents

B.4 Strings 283
B.5 Numbers 283
B.6 Tuples 284
B.7 Maybe 284
B.8 Lists 285
B.9 Functions 287
B.10 Input/output 287
B.11 Functors 288
B.12 Applicatives 289
B.13 Monads 290
B.14 Alternatives 290
B.15 MonadPlus 291
B.16 Monoids 292
B.17 Foldables 294
B.18 Traversables 297

Bibliography 298
Index 300