The Haskell School of Expression

LEARNING FUNCTIONAL PROGRAMMING THROUGH MULTIMEDIA

PAUL HUDAK

Contents

Preface *page* xiii

1 Problem Solving, Programming, and Calculation 1
- 1.1 Computation by Calculation in Haskell 2
- 1.2 Expressions, Values, and Types 6
- 1.3 Function Types and Type Signatures 8
- 1.4 Abstraction, Abstraction, Abstraction 10
  - 1.4.1 Naming 10
  - 1.4.2 Functional Abstraction 12
  - 1.4.3 Data Abstraction 13
- 1.5 Code Reuse and Modularity 17
- 1.6 Beware of Programming with Numbers 17

2 A Module of Shapes: Part I 21
- 2.1 Geometric Shapes 22
- 2.2 Areas of Shapes 25
- 2.3 Cleaning Up 34

3 Simple Graphics 35
- 3.1 Basic Input/Output 36
- 3.2 Graphics Windows 40
- 3.3 Drawing Graphics Other Than Text 42
- 3.4 Some Examples 43

4 Shapes II: Drawing Shapes 48
- 4.1 Dealing With Different Coordinate Systems 48
- 4.2 Converting Shapes to Graphics 51
- 4.3 Some Examples 52
- 4.4 In Retrospect 54

5 Polymorphic and Higher-Order Functions 56
- 5.1 Polymorphic Types 56

5.2 Abstraction Over Recursive Definitions 58
5.2.1 Map is Polymorphic 59
5.2.2 Using *map* 60
5.3 Append 63
5.3.1 The Efficiency and Fixity of Append 64
5.4 Fold 65
5.4.1 Haskell's Folds 67
5.4.2 Why Two Folds? 68
5.5 A Final Example: Reverse 69
5.6 Errors 71

6 Shapes III: Perimeters of Shapes 74
6.1 Perimeters of Shapes 74

7 Trees 81
7.1 A Tree Data Type 81
7.2 Operations on Trees 83
7.3 Arithmetic Expressions 84

8 A Module of Regions 87
8.1 The Region Data Type 87
8.2 The Meaning of Shapes and Regions 91
8.2.1 The Meaning of Shapes 92
8.2.2 The Encoding of the Meaning of Shapes 93
8.2.3 The Meaning of Regions 96
8.2.4 The Encoding of the Meaning of Regions 97
8.3 Algebraic Properties of Regions 99
8.4 In Retrospect 101

9 More About Higher-Order Functions 105
9.1 Currying 105
9.2 Sections 108
9.3 Anonymous Functions 110
9.4 Function Composition 111

10 Drawing Regions 114
10.1 The Picture Data Type 115
10.2 Drawing Pictures 115
10.3 Drawing Regions 116
10.3.1 From Regions to Graphics Regions: First Attempt 117
10.3.2 From Regions to Graphics Regions: Second Attempt 119
10.3.3 Translating Shapes into Graphics Regions 121
10.3.4 Examples 123

10.4 User Interaction 126
10.5 Putting it all Together 127
10.5.1 Examples 128

11 **Proof by Induction** 131
11.1 Induction and Recursion 131
11.2 Examples of List Induction 132
11.2.1 Proving Function Equivalences 133
11.3 Useful Properties on Lists 137
11.3.1 Function Strictness 137
11.4 Induction on Other Data Types 141
11.4.1 A More Efficient Exponentiation Function 143

12 **Qualified Types** 147
12.1 Equality 148
12.2 Defining Your Own Type Classes 150
12.3 Inheritance 153
12.4 Haskell's Standard Type Classes 154
12.5 Derived Instances 157
12.6 Reasoning With Type Classes 160

13 **A Module of Simple Animations** 163
13.1 What is an Animation? 163
13.2 Representing an Animation 165
13.3 An Animator 167
13.4 Fun With Type Classes 172
13.4.1 Rising to the Level of Animations 172
13.4.2 Type Classes to the Rescue 172
13.4.3 Defining New Type Classes for Behaviors 176
13.5 Lifting to the Limit 177
13.6 Time Transformation 179
13.7 A Final Example: A Kaleidoscope Program 180

14 **Programming With Streams** 187
14.1 Lazy Evaluation 187
14.2 Recursive Streams 190
14.3 Stream Diagrams 193
14.4 Lazy Patterns 195
14.5 Memoization 198
14.6 Inductive Properties of Infinite Lists 201

15 **A Module of Reactive Animations** 208
15.1 FAL by Example 209

15.1.1 Basic Reactivity 209
15.1.2 Event Choice 210
15.1.3 Recursive Event Processing 211
15.1.4 Events with Data 212
15.1.5 Snapshot 212
15.1.6 Boolean Events 212
15.1.7 Integration 213
15.2 Implementing FAL 214
15.2.1 An Implementation Strategy 215
15.2.2 Incremental Sampling 217
15.2.3 Final Refinements 219
15.2.4 Representing Events 220
15.3 The Implementation 220
15.3.1 Behaviors 221
15.3.2 Events 225
15.3.3 An Example 228
15.4 Extensions 229
15.4.1 Variations on *switch* 231
15.4.2 Mouse Motion 232
15.5 Paddleball in Twenty Lines 233

16 Communicating With the Outside World 236
16.1 Files, Channels, and Handles 236
16.1.1 Why Use Handles? 238
16.1.2 Channels 238
16.2 Exception Handling 239
16.3 First-Class Channels and Concurrency 242

17 Rendering Reactive Animations 245
17.1 Preliminaries 245
17.2 Reactimate 246
17.3 Window User 247

18 Higher-Order Types 249
18.1 The Functor Class 249
18.2 The Monad Class 251
18.2.1 Other Instances of Monad 255
18.2.2 Other Monadic Operations 259
18.3 The MonadPlus Class 259
18.4 State Monads 261
18.5 Type Class Type Errors 263

19 An Imperative Robot Language 265
19.1 IRL by Example 266
19.2 *Robot* is a State Monad 270
19.3 The Implementation of IRL Commands 272
19.3.1 Robot Orientation 273
19.3.2 Using the Pen 274
19.3.3 Playing With Coins 274
19.3.4 Logic and Control 274
19.4 All the World is a Grid 276
19.5 Robot Graphics 281
19.6 Putting it all Together 282

20 Functional Music Composition 287
20.1 The Music Data Type 288
20.2 Higher-Level Constructions 293
20.2.1 Lines and Chords 293
20.2.2 Delay and Repeat 293
20.2.3 Polyrhythms 294
20.2.4 Determining Duration 295
20.2.5 Reversing Musical Structure 295
20.2.6 Truncating Parallel Composition 296
20.2.7 Trills 297
20.2.8 Percussion 298
20.3 A Couple of Final Examples 300
20.3.1 Cascades 300
20.3.2 Self-Similar (Fractal) Music 301

21 Interpreting Functional Music 304
21.1 Interpreting Music: A *Performance* 305
21.2 An Algebra of Music 308

22 From Performance to MIDI 313
22.1 An Introduction to MIDI 313
22.2 The Conversion Process 314
22.3 Putting It All Together 319

23 A Tour of the PreludeList Module 321
23.1 The PreludeList Module 321
23.2 Simple List Selector Functions 322
23.3 Index-Based Selector Functions 323
23.4 Predicate-Based Selector Functions 324
23.5 Fold-like Functions 325
23.6 List Generators 327

	23.7	String-Based Functions	327
	23.8	Boolean List Functions	328
	23.9	List Membership Functions	329
	23.10	Arithmetic on Lists	329
	23.11	List Combining Functions	330
24	**A Tour of Haskell's Standard Type Classes**		332
	24.1	The Ordered Class	332
	24.2	The Enumeration Class	333
	24.3	The Bounded Class	334
	24.4	The Show Class	334
	24.5	The Read Class	338
	24.6	The Index Class	341
	24.7	The Numeric Classes	343
A	**Built-in Types Are Not Special**		345
B	**Pattern-Matching Details**		348
	Bibliography		353
	Index		357