

Contents

- 1. Introduction** 1
 - 1.1. New Tools 1
 - 1.2. New Techniques 3
 - 1.3. A New Approach 4
- 2. Welcome to Lisp** 7
 - 2.1. Form 7
 - 2.2. Evaluation 9
 - 2.3. Data 10
 - 2.4. List Operations 12
 - 2.5. Truth 13
 - 2.6. Functions 14
 - 2.7. Recursion 16
 - 2.8. Reading Lisp 17
 - 2.9. Input and Output 18
 - 2.10. Variables 19
 - 2.11. Assignment 21
 - 2.12. Functional Programming 22
 - 2.13. Iteration 23
 - 2.14. Functions as Objects 25
 - 2.15. Types 27
 - 2.16. Looking Forward 27
- 3. Lists** 31
 - 3.1. Conses 31
 - 3.2. Equality 34
 - 3.3. Why Lisp Has No Pointers 34
 - 3.4. Building Lists 36
 - 3.5. Example: Compression 36
 - 3.6. Access 39
 - 3.7. Mapping Functions 40
 - 3.8. Trees 40
 - 3.9. Understanding Recursion 42
 - 3.10. Sets 43
 - 3.11. Sequences 45
 - 3.12. Stacks 47
 - 3.13. Dotted Lists 49
 - 3.14. Assoc-lists 51
 - 3.15. Example: Shortest Path 51
 - 3.16. Garbage 54
- 4. Specialized Data Structures** 58
 - 4.1. Arrays 58
 - 4.2. Example: Binary Search 60
 - 4.3. Strings and Characters 61
 - 4.4. Sequences 63
 - 4.5. Example: Parsing Dates 66
 - 4.6. Structures 69
 - 4.7. Example: Binary Search Trees 71
 - 4.8. Hash Tables 76
- 5. Control** 81
 - 5.1. Blocks 81
 - 5.2. Context 83
 - 5.3. Conditionals 85
 - 5.4. Iteration 87
 - 5.5. Multiple Values 89
 - 5.6. Aborts 91

5.7. Example: Date Arithmetic 92

6. Functions 99

- 6.1. Global Functions 99
- 6.2. Local Functions 101
- 6.3. Parameter Lists 102
- 6.4. Example: Utilities 104
- 6.5. Closures 107
- 6.6. Example: Function Builders 109
- 6.7. Dynamic Scope 112
- 6.8. Compilation 113
- 6.9. Using Recursion 114

7. Input and Output 119

- 7.1. Streams 119
- 7.2. Input 121
- 7.3. Output 123
- 7.4. Example: String Substitution 125
- 7.5. Macro Characters 130

8. Symbols 133

- 8.1. Symbol Names 133
- 8.2. Property Lists 134
- 8.3. Symbols Are Big 135
- 8.4. Creating Symbols 136
- 8.5. Multiple Packages 136
- 8.6. Keywords 137
- 8.7. Symbols and Variables 138
- 8.8. Example: Random Text 138

9. Numbers 143

- 9.1. Types 143
- 9.2. Conversion and Extraction 144
- 9.3. Comparison 146
- 9.4. Arithmetic 147
- 9.5. Exponentiation 148
- 9.6. Trigonometric Functions 149
- 9.7. Representation 150
- 9.8. Example: Ray-Tracing 151

10. Macros 160

- 10.1. Eval 160
- 10.2. Macros 162
- 10.3. Backquote 163
- 10.4. Example: Quicksort 164
- 10.5. Macro Design 165
- 10.6. Generalized Reference 168
- 10.7. Example: Macro Utilities 169
- 10.8. On Lisp 173

11. CLOS 176

- 11.1. Object-Oriented Programming 176
- 11.2. Classes and Instances 179
- 11.3. Slot Properties 179
- 11.4. Superclasses 181
- 11.5. Precedence 182
- 11.6. Generic Functions 184
- 11.7. Auxiliary Methods 187
- 11.8. Method Combination 189
- 11.9. Encapsulation 190
- 11.10. Two Models 192

12. Structure 195

- 12.1. Shared Structure 195
- 12.2. Modification 198
- 12.3. Example: Queues 200
- 12.4. Destructive Functions 201
- 12.5. Example: Binary Search Trees 203
- 12.6. Example: Doubly-Linked Lists 204
- 12.7. Circular Structure 208
- 12.8. Constant Structure 210

13. Speed 213

- 13.1. The Bottleneck Rule 213
- 13.2. Compilation 214
- 13.3. Type Declarations 217
- 13.4. Garbage Avoidance 222
- 13.5. Example: Pools 226
- 13.6. Fast Operators 228
- 13.7. Two-Phase Development 229

14. Advanced Topics 232

- 14.1. Type Specifiers 232
- 14.2. Binary Streams 234
- 14.3. Read-Macros 235
- 14.4. Packages 236
- 14.5. The Loop Facility 239
- 14.6. Conditions 244

15. Example: Inference 247

- 15.1. The Aim 247
- 15.2. Matching 248
- 15.3. Answering Queries 251
- 15.4. Analysis 255

16. Example: Generating HTML 257

- 16.1. HTML 257
- 16.2. HTML Utilities 259
- 16.3. An Iteration Utility 262
- 16.4. Generating Pages 264

17. Example: Objects 269

- 17.1. Inheritance 269
- 17.2. Multiple Inheritance 271
- 17.3. Defining Objects 273
- 17.4. Functional Syntax 274
- 17.5. Defining Methods 275
- 17.6. Instances 277
- 17.7. New Implementation 277
- 17.8. Analysis 284

A. Debugging 287

B. Lisp in Lisp 295

C. Changes to Common Lisp 304

D. Language Reference 310

Notes 401

Index 415