AI Algorithms, Data Structures, and Idioms in Prolog, Lisp, and Java

George F Luger
William A Stubblefield

2009 Pearson Education
Contents

Preface ix

Part I Language Idioms and the Master Programmer 1

Chapter 1 Idioms, Patterns, and Programming 3
1.1 Introduction: Idioms and Patterns 3
1.2 Selected Examples of Language Idioms 6
1.3 A Brief History of Three Programming Paradigms 11
1.4 A Summary of Our Task 15

Part II Programming in Prolog 17

Chapter 2 Prolog: Representation 19
2.1 Introduction: Logic-Based Representation 19
2.2 Prolog Syntax 20
2.3 Creating, Changing, and Tracing a Prolog Computation 24
2.4 Lists and Recursion in Prolog 25
2.5 Structured Representation and Inheritance Search 28
Exercises 32

Chapter 3 Abstract Data Types and Search 33
3.1 Introduction 33
3.2 Using cut to Control Search in Prolog 36
3.3 Abstract Data Types (ADTs) in Prolog 38
Exercises 42

Chapter 4 Depth-, Breadth-, and Best-First Search 43
4.1 Production System Search in Prolog 43
4.2 A Production System Solution of the FWGC Problem 46
4.3 Designing Alternative Search Strategies 52
Exercises 58

Chapter 5 Meta-Linguistic Abstraction, Types, and Meta-Interpreters 59
5.1 Meta-Interpreters, Types, and Unification 59
5.2 Types in Prolog 61
5.3 Unification, Variable Binding, and Evaluation 64
Exercises 68
Chapter 6 Three Meta-Interpreters: Prolog in Prolog, EXSHELL, and a Planner 59
  6.1 An Introduction to Meta-Interpreters: Prolog in Prolog 69
  6.2 A Shell for a Rule-Based System 73
  6.3 A Prolog Planner 82
  Exercises 85

Chapter 7 Machine Learning Algorithms in Prolog 87
  7.1 Machine Learning: Version Space Search 87
  7.2 Explanation Based Learning in Prolog 100
  Exercises 106

Chapter 8 Natural Language Processing in Prolog 107
  8.1 Natural Language Understanding 107
  8.2 Prolog Based Semantic Representation 108
  8.3 A Context-Free Parser in Prolog 111
  8.4 Probabilistic Parsers in Prolog 114
  8.5 A Context-Sensitive Parser in Prolog 119
  8.6 A Recursive Descent Semantic Net Parser 120
  Exercises 123

Chapter 9 Dynamic Programming and the Earley Parser 125
  9.1 Dynamic Programming Revisited 125
  9.2 The Earley Parser 126
  9.3 The Earley Parser in Prolog 134
  Exercises 139

Chapter 10 Prolog: Final Thoughts 141
  10.1 Towards a Procedural Semantics 141
  10.2 Prolog and Automated Reasoning 144
  10.3 Prolog Idioms, Extensions, and References 145

Part III Programming in Lisp 149

Chapter 11 S-Expressions, the Syntax of Lisp 151
  11.1 Introduction to Symbol Expressions 151
  11.2 Control of Lisp Evaluation 154
  11.3 Programming in Lisp: Creating New Functions 156
  11.4 Program Control: Conditionals and Predicates 157
  Exercises 160
Contents

Exercises 266

Chapter 20  Lisp: Final Thoughts 267

Part IV  Programming in Java 269

Chapter 21  Java, Representation and Object-Oriented Programming 273
21.1 Introduction to O-O Representation and Design 273
21.2 Object Orientation 274
21.3 Classes and Encapsulation 275
21.4 Polymorphism 276
21.5 Inheritance 277
21.6 Interfaces 280
21.7 Scoping and Access 282
21.8 The Java Standard Library 283
21.9 Conclusions: Design in Java 284
Exercises 285

Chapter 22  Problem Spaces and Search 287
21.1 Abstraction and Generality in Java 287
21.2 Search Algorithms 288
21.3 Abstracting Problem States 292
21.4 Traversing the-Solution Space 295
21.5 Putting the Framework to Use 298
Exercises 303

Chapter 23  Java Representation for Predicate Calculus and Unification 305
23.1 Introduction to the Task 305
23.2 A Review of the Predicate Calculus and Unification 307
23.3 Building a Predicate Calculus Problem Solver in Java 310
23.4 Design Discussion 320
23.5 Conclusions: Mapping Logic into Objects 322
Exercises 323

Chapter 24  A Logic-Based Reasoning System 325
24.1 Introduction 325
24.2 Reasoning in Logic as Searching an And/Or Graph 325
24.3 The Design of a Logic-Based Reasoning System 329
24.4 Implementing Complex Logic Expressions 330
24.5 Logic-Based Reasoning as And/Or Graph Search 335
24.6 Testing the Reasoning System 346
Chapter 25  **An Expert System Shell**  351  
25.1 Introduction: Expert Systems  351  
25.2 Certainty Factors and the Unification Problem Solver  352  
25.3 Adding User Interactions  358  
25.4 Design Discussion  360  
*Exercises*  361  

Chapter 26  **Case Studies: JESS and other Expert System Shells in Java**  363  
26.1 Introduction  363  
26.2 JESS  363  
26.3 Other Expert System Shells  364  
26.4 Using Open Source Tools  365  

Chapter 27  **ID3: Learning from Examples**  367  
27.1 Introduction to Supervised Learning  367  
27.2 Representing Knowledge as Decision Trees  367  
27.3 A Decision Tree Induction Program  370  
27.4 ID3: An Information Theoretic Tree Induction Algorithm  385  
*Exercises*  388  

Chapter 28  **Genetic and Evolutionary Computing**  389  
28.1 Introduction  389  
28.2 The Genetic Algorithm: A First Pass  389  
28.3 A GA Java Implementation in Java  393  
28.4 Conclusion: Complex Problem Solving and Adaptation  401  
*Exercises*  401  

Chapter 29  **Case Studies: Java Machine Learning Software Available on the Web**  403  
29.1 Java Machine Learning Software  403  

Chapter 30  **The Earley Parser: Dynamic Programming in Java**  405  
30.1 Chart Parsing  405  
30.2 The Earley Parser: Components  406  
30.3 The Earley Parser: Java Code  408  
30.4 The Completed Parser  414  
30.5 Generating Parse Trees from Charts and Grammar Rules  419  
*Exercises*  422
### Contents

**Chapter 31**  
**Case Studies: Java Natural Language Tools on the Web**  
31.1 Java Natural Language Processing Software 423  
31.2 LingPipe from the University of Pennsylvania 423  
31.3 The Stanford Natural Language Processing Group Software 425  
31.4 Sun's Speech API 426

**Part V**  
**Model Building and the Master Programmer** 429

**Chapter 32**  
**Conclusion: The Master Programmer** 431  
32.1 Paradigm-Based Abstractions and Idioms 431  
32.2 Programming as a Tool for Exploring Problem Domains 433  
32.3 Programming as a Social Activity 434  
32.4 Final Thoughts 437

**Bibliography** 439  
**Index** 443