ARTIFICIAL INTELLIGENCE
and the Design of Expert Systems

George F. Luger
William A. Stubblefield

1st edition 1989 Benjamin - Cummings

CONTENTS

Preface vii

PART I
ARTIFICIAL INTELLIGENCE, ITS ROOTS AND SCOPE 1

Artificial Intelligence—An Attempted Definition 1

1 AI: HISTORY AND APPLICATIONS 3

1.1 From Eden to ENIAC: Attitudes Toward Intelligence, Knowledge, and Human Artifice 3

1.2 Overview of AI Application Areas 13

1.2.1 Game Playing 14

1.2.2 Automated Reasoning and Theorem Proving 15

1.2.3 Expert Systems 16

1.2.4 Natural Language Understanding and Semantic Modeling 18

1.2.5 Modeling Human Performance 19

1.2.6 Planning and Robotics 20

1.2.7 Languages and Environments for AI 20

1.2.8 Machine Learning 21

1.3 Artificial Intelligence—A Summary 22

1.4 Epilogue and References 23

1.5 Exercises 24

PART II

ARTIFICIAL INTELLIGENCE AS REPRESENTATION AND SEARCH 27

Knowledge Representation 28

A. Handle qualitative knowledge. 31

B. Allow new knowledge to be inferred from a basic set of facts. 32

C. Allow representation of general principles as well as specific situations. 32

D. Capture complex semantic meaning. 32

E. Allow for meta-level reasoning. 34

Problem Solving as Search 35

2 THE PREDICATE CALCULUS 41

2.0 Introduction 41

2.1 The Propositional Calculus (Optional Review) 41

2.1.1 Symbols and Sentences 41

2.1.2 The Semantics of Propositional Calculus 43

2.2 The Predicate Calculus 46

2.2.1 The Syntax of Predicates and Sentences 46

2.2.2 A Semantics for the Predicate Calculus 53

2.3 Using Inference Rules to Produce Predicate Calculus Expressions 58

2.3.1 Inference Rules 58

2.3.2 Unification 62

2.3.3 A Unification Example 66

2.4 Application: A Logic-Based Financial Advisor 68

2.5 Epilogue and References 73

2.6 Exercises 74

3 STRUCTURES AND STRATEGIES FOR STATE SPACE SEARCH 77

3.0 Introduction 77

3.1 Graph Theory (Optional Review) 80

3.1.1 Structures for State Space Search 80

3.1.2 State Space Representation of Problems 82

3.2 Strategies for State Space Search 88

3.2.1 Data-Driven and Goal-Driven Search 88

3.2.2 Implementing Graph Search 90

3.2.3 Depth- and Breadth-First Search 94

3.2.4 Depth-First Search with Iterative Deepening 101

3.3 Using the State Space to Represent Reasoning with the Predicate Calculus 102

3.3.1 State Space Description of a Logical System 102

3.3.2 And/Or Graphs 103

3.3.3 Further Examples and Applications 106

3.4 Epilogue and References 115

3.5 Exercises 116

4 CONTROL AND IMPLEMENTATION OF STATE SPACE SEARCH 119

4.0 Introduction 119

4.1 Recursion-Based Search 120

4.1.1 Recursion (Optional Review) 120

4.1.2 Recursive Search 121

4.2 Pattern-Directed Search 123

4.3 Production Systems 130

4.3.1 Definition and History 130

4.3.2 Examples of Production Systems 134

4.3.3 Control of Search in Production Systems 140

4.3.4 Advantages of Production Systems for AI 144

4.4 Epilogue and References 146

4.5 Exercises 146

5 HEURISTIC SEARCH 149

5.0 Introduction 149

5.1 An Algorithm for Heuristic Search 153

5.1.1 Implementing "Best-First" Search 153

5.1.2 Implementing Heuristic Evaluation Functions 156

5.1.3 Heuristic Search and Expert Systems 163

6.3 PROLOG: Toward a Nonprocedural Computing Language 225

6.4 Epilogue and References 231

6.5 Exercises 232

7 **LISP 235**

7.0 Introduction 235

7.1 LISP: A Brief Overview (Optional) 236

7.1.1 Symbolic Expressions, the Syntactic Basis of LISP 236

7.1.2 Programming in LISP: Creating New Functions 240

7.1.3 Program Control in LISP: Conditionals and Predicates 241

7.1.4 Functions, Lists, and Symbolic Computing 244

7.1.5 Control of LISP Evaluation: quote and eval 246

7.1.6 Lists as Recursive Structures 247

7.1.7 Nested Lists, Structure, and car/cdr Recursion 250

7.1.8 Functional Programming, Side Effects, setq, and let 253

7.2 Search Algorithms in LISP: A Functional Approach to the Farmer, Wolf, Goat, and Cabbage Problem 257

7.3 Search Strategies in LISP 262

7.3.1 Breadth-First Search 262

7.3.2 Depth-First and Best-First Search 264

7.3.3 A General Graph Search Algorithm 265

7.4 Higher-Order Functions and Procedural Abstraction 267

7.4.1 Maps and Filters 267

7.4.2 Functional Arguments and Lambda Expressions 269

7.5 Pattern Matching in LISP 270

7.6 A Recursive Unification Function 273

7.6.1 Implementing the Unification Algorithm 273

7.6.2 Implementing Substitution Sets Using Association Lists 275

7.7 Interpreters and Embedded Languages 277

7.8 Epilogue and References 279

7.9 Exercises 280

PART IV
REPRESENTATIONS FOR KNOWLEDGE-BASED SYSTEMS 285

8 RULE-BASED EXPERT SYSTEMS 291

8.0 Introduction 291

8.1 Overview of Expert System Technology 293

- 8.1.1 Design of Rule-Based Expert Systems 293
- 8.1.2 Selecting a Problem for Expert System Development 296
- 8.1.3 Overview of "Knowledge Engineering" 297

8.2 A Framework for Organizing and Applying Human Knowledge 301

- 8.2.1 Production Systems, Rules, and the Expert System Architecture 301
- 8.2.2 Explanation and Transparency 306
- 8.2.3 Heuristics and Control in Expert Systems 307

8.3 Using Uncertainty Measures in Expert Systems 308

- 8.3.1 Introduction 308
- 8.3.2 Bayesian Probability Theory 309
- 8.3.3 A Theory for Certainty 311
- 8.3.4 Other Approaches to Uncertainty 313

8.4 MYCIN, a Case Study 315

- 8.4.1 Introduction 315
- 8.4.2 Representation of Rules and Facts 316
- 8.4.3 MYCIN Diagnosing an Illness 318
- 8.4.4 Evaluation of Expert Systems 324
- 8.4.5 Knowledge Acquisition and the Teiresias Knowledge Base Editor 326

8.5 Epilogue and References 331

8.6 Exercises 331

9 KNOWLEDGE REPRESENTATION 333

9.0 Knowledge Representation Languages 333

9.1 Issues in Knowledge Representation 335

9.2 A Survey of Network Representations 336

- 9.2.1 Associationist Theories of Meaning 336

9.2.2 Early Work in Semantic Nets 341
9.2.3 Standardization of Network Relationships 343

9.3 A Network Representation Language 349

9.3.1 Conceptual Graphs 349
9.3.2 Types, Individuals, and Names 351
9.3.3 The Type Hierarchy 353
9.3.4 Generalization and Specialization of Conceptual Graphs 354
9.3.5 Propositional Nodes 356
9.3.6 Conceptual Graphs and Logic 357

9.4 Structured Representations 359

9.4.1 Frames 359
9.4.2 Scripts 363

9.5 Type Hierarchies, Inheritance, and Exception Handling 367

9.6 Further Problems in Knowledge Representation 371

9.7 Epilogue and References 372

9.8 Exercises 374

10 NATURAL LANGUAGE 377

10.0 Role of Knowledge in Language Understanding 377

10.1 The Natural Language Problem 379

10.1.1 Levels of Analysis 379
10.1.2 Stages of Natural Language Analysis 380

10.2 Syntax 382

10.2.1 Specification and Parsing Using Context-Free Grammars 382
10.2.2 Transition Network Parsers 384
10.2.3 The Chomsky Hierarchy and Context-Sensitive Grammars 388

10.3 Combining Syntax and Semantics in ATN Parsers 391

10.3.1 Augmented Transition Network Parsers 391
10.3.2 Combining Syntax and Semantics 396

10.4 Natural Language Applications 401

10.4.1 Story Understanding and Question Answering 401
10.4.2 A Data Base Front End 401

10.5 Epilogue and References 405

10.6 Exercises 407

11 AUTOMATED REASONING 409

11.0 Introduction to Weak Methods in Theorem Proving 409

11.1 The General Problem Solver and Difference Tables 410

11.2 Resolution Theorem Proving 415

11.2.1 Introduction 415
11.2.2 Producing the Clause Form for Resolution Refutations 418
11.2.3 The Resolution Proof Procedure 421
11.2.4 Strategies and Simplification Techniques for Resolution 425
11.2.5 Answer Extraction from Resolution Refutations 431

11.3 Further Issues in the Design of Automated Reasoning Programs 434

11.3.1 Uniform Representations for Weak Method Solutions 435
11.3.2 Alternative Inference Rules 438
11.3.3 Search Strategies and Their Use 440

11.4 Epilogue and References 441

11.5 Exercises 442

PART V
ADVANCED AI PROGRAMMING TECHNIQUES 445

AI Languages and Meta-Interpreters 445

Object-Oriented Programming 446

Hybrid Environments 447

12 ADVANCED REPRESENTATION IN PROLOG 449

12.0 Introduction 449

12.1 PROLOG Tools: Meta-Predicates, Types, and Unification 450

12.1.1 Meta-Logical Predicates 450
12.1.2 Types in PROLOG 451
12.1.3 Unification, the Engine for Predicate Matching and Evaluation 454

12.2 Advanced Representations in PROLOG 458

12.2.1 Shell for a Rule-Based Expert System 460
12.2.2 Semantic Nets in PROLOG 469
12.2.3 Frames and Schemata in PROLOG 471
12.2.4 Natural Language Understanding in PROLOG 474

12.3 PROLOG and Programming with Logic 475

12.3.1 Introduction 475

12.3.2 Logic Programming and PROLOG 476

12.3.3 Semantics of Flat Concurrent PROLOG (E. Shapiro 1987) 482

12.4 Epilogue and References 484

12.5 Exercises 485

13 ADVANCED LISP PROGRAMMING TECHNIQUES FOR ARTIFICIAL INTELLIGENCE 487

13.0 Introduction: Abstraction and Complexity 487

13.1 Logic Programming in LISP 488

13.1.1 A Simple Logic Programming Language 488

13.1.2 Streams and Stream Processing 490

13.1.3 A Stream-Based Logic Programming Interpreter 493

13.2 Streams and Delayed Evaluation 497

13.3 An Expert System Shell in LISP 500

13.3.1 Implementing Certainty Factors 501

13.3.2 Architecture of lisp-shell 502

13.3.3 User Queries and Working Memory 505

13.3.4 Small Classification System Using lisp-shell 506

13.4 Network Representations and Inheritance 508

13.4.1 Representing Semantic Nets in LISP 508

13.4.2 Implementing Inheritance 511

13.5 Epilogue and References 512

13.6 Exercises 513

14 OBJECTS, MESSAGES, AND HYBRID EXPERT SYSTEM DESIGN 515

14.0 Introduction 515

14.1 Object-Oriented Knowledge Representation 516

14.1.1 Using Objects for Inheritance and Abstraction 516

14.1.2 Object-Oriented Knowledge Base 524

14.2 LISP and Object-Oriented Programming 525

14.2.1 OOPS: A Simple Object-Oriented Programming Language 526
14.2.2 Implementing OOPS in LISP 528
14.2.3 An Object-Oriented Simulation Using OOPS 532
14.2.4 Evaluating OOPS 536

14.3 Object-Oriented Programming and Concurrency in PROLOG 537

14.3.1 Introduction 537
14.3.2 Objects in Concurrent PROLOG (E. Shapiro 1987) 538

14.4 Hybrid Expert System Tools 545

14.4.1 Hybrid Environments: Integrating Objects, Rules, and Procedures 545
14.4.2 Facets, Demons, and Graphics in Hybrid Environments 546

14.5 Epilogue and References 548

14.6 Exercises 549

15 ADVANCED TOPICS IN AI PROBLEM SOLVING 551

15.0 Introduction 551

15.1 Planning, STRIPS, and Triangle Tables 552

15.2 Blackboard Architecture for Problem Solving 561

15.3 Learning 563

15.3.1 Varieties of Training Data and Learning Tasks 568
15.3.2 Representations and Meta-Operations 569
15.3.3 Search Strategies and Heuristics in Learning Systems 570
15.3.4 Other Approaches 575

15.4 Cognitive Science 576

15.4.1 Overview and Application Areas 576
15.4.2 Cognitive Science: Open Issues 580

15.5 Neural Networks 581

15.5.1 Early Work in Neural Computing 582
15.5.2 Current Work in Neural Nets 584
15.5.3 Neural Nets: Problems and Promise 584

15.6 Epilogue and References 586

15.7 Exercises 587

PART VI
EPILOGUE 591

16 ARTIFICIAL INTELLIGENCE AS EMPIRICAL INQUIRY 593

16.1 Artificial Intelligence: A Final Definition 593

16.2 Expert Systems and Future Knowledge Media 598

16.3 Representational Models for Intelligence: Current Limitations 600

16.4 References 602

APPENDIX 1
PROLOG PROGRAMS 603

A1.1 exshell and cars 603

A1.2 Interpreter for Concurrent PROLOG 613

APPENDIX 2
LISP PROGRAMS 619

A2.1 Stream Functions with Delayed Evaluation 619

A2.2 Logic Programming Interpreter 620

A2.3 LISP-Based Expert System Shell 623

A2.4 OOPS 628

Bibliography 631

Author Index 645

Subject Index 649

Illustration and Text Credits 659