ARTIFICIAL INTELLIGENCE
STRUCTURES AND STRATEGIES FOR COMPLEX PROBLEM SOLVING
SECOND EDITION

1973 Benjamin Cummings
TABLE OF 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
  1.1.1 Historical Foundations 4
  1.1.2 The Development of Logic 7
  1.1.3 The Turing Test 10

1.2 Overview of AI Application Areas 13
  1.2.1 Game Playing 14
  1.2.2 Automated Reasoning and Theorem Proving 14
  1.2.3 Expert Systems 15
  1.2.4 Natural Language Understanding and Semantic Modeling 17
  1.2.5 Modeling Human Performance 18
  1.2.6 Planning and Robotics 19
  1.2.7 Languages and Environments for AI 20
  1.2.8 Machine Learning 20
  1.2.9 Neural Networks or Parallel Distributed Processing (PDP) 21
  1.2.10 AI and Philosophy 22
5.2 Pattern-Directed Search  156
5.3 Production Systems  163
5.3.1 Definition and History  163
5.3.2 Examples of Production Systems  167
5.3.3 Control of Search in Production Systems  172
5.3.4 Advantages of Production Systems for AI  177
5.4 Predicate Calculus and Planning  179
5.5 The Blackboard Architecture for Problem Solving  187
5.6 Epilogue and References  190
5.7 Exercises  191

PART III
LANGUAGES FOR AI PROBLEM SOLVING  195

Languages, Understanding, and Levels of Abstraction  196
Requirements for AI Languages  198
The Primary AI Languages: LISP and PROLOG  205
  PROLOG  206
  LISP  207
Selecting an Implementation Language  208

6  AN INTRODUCTION TO PROLOG  210
6.0 Introduction  210
6.1 Syntax for Predicate Calculus Programming  211
  6.1.1 Representing Facts and Rules  211
  6.1.2 Creating, Changing, and Monitoring the PROLOG Environment  215
  6.1.3 Recursion-Based Search in PROLOG  216
  6.1.4 Recursive Search in PROLOG  219
  6.1.5 The Use of Cut to Control Search in PROLOG  222
6.2 Abstract Data Types (ADTs) in PROLOG  224
  6.2.1 The ADT Stack  224
  6.2.2 The ADT Queue  225
  6.2.3 The ADT Priority Queue  226
  6.2.4 The ADT Set  227
PART IV
REPRESENTATIONS FOR KNOWLEDGE-BASED SYSTEMS 303

8 RULE-BASED EXPERT SYSTEMS 308

8.0 Introduction 308

8.1 Overview of Expert Systems Technology 310
  8.1.1 Design of Rule-Based Expert Systems 310
  8.1.2 Selecting a Problem for Expert System Development 312
  8.1.3 The Knowledge Engineering Process 314
  8.1.4 Conceptual Models and Their Role in Knowledge Acquisition 317

8.2 A Framework for Organizing and Applying Human Knowledge 320
  8.2.1 Production Systems, Rules, and the Expert System Architecture 320
  8.2.2 Explanation and Transparency 323
  8.2.3 Heuristics and Control in Expert Systems 326

8.3 Managing Uncertainty in Expert Systems 326
  8.3.1 Introduction 326
  8.3.2 Bayesian Probability Theory 328
  8.3.3 The Stanford Certainty Factor Algebra 329
  8.3.4 Nonmonotonic Logic and Reasoning with Beliefs 332
  8.3.5 Fuzzy Logic, Dempster/Shafer, and Other Approaches to Uncertainty 333

8.4 MYCIN: A Case Study 334
  8.4.1 Introduction 334
  8.4.2 Representation of Rules and Facts 335
  8.4.3 MYCIN Diagnosing an Illness 337
  8.4.4 Evaluation of Expert Systems 342
  8.4.5 Knowledge Acquisition and the Telesias Knowledge-Base Editor 344

8.5 Epilogue and References 349

8.6 Exercises 350
9 KNOWLEDGE REPRESENTATION 352

9.0 Knowledge Representation Languages 352
9.1 Issues in Knowledge Representation 354
9.2 A Survey of Network Representations 356
  9.2.1 Associationist Theories of Meaning 356
  9.2.2 Early Work in Semantic Nets 360
  9.2.3 Standardization of Network Relationships 362
9.3 Conceptual Graphs: A Network Representation Language 368
  9.3.1 Introduction to Conceptual Graphs 368
  9.3.2 Types, Individuals, and Names 369
  9.3.3 The Type Hierarchy 372
  9.3.4 Generalization and Specialization 372
  9.3.5 Propositional Nodes 376
  9.3.6 Conceptual Graphs and Logic 377
9.4 Structured Representations 378
  9.4.1 Frames 378
  9.4.2 Scripts 383
9.5 Type Hierarchies, Inheritance, and Exception Handling 386
9.6 Further Problems in Knowledge Representation 389
9.7 Epilogue and References 392
9.8 Exercises 393

10 NATURAL LANGUAGE 396

10.0 Role of Knowledge in Language Understanding 396
10.1 The Natural Language Problem 398
  10.1.1 Introduction 398
  10.1.2 Stages of Language Analysis 399
10.2 Syntax 401
  10.2.1 Specification and Parsing Using Context-Free Grammars 401
  10.2.2 Transition Network Parsers 403
  10.2.3 The Chomsky Hierarchy and Context-Sensitive Grammars 408
10.3 Combining Syntax and Semantics in ATN Parsers 410
  10.3.1 Augmented Transition Network Parsers 411
  10.3.2 Combining Syntax and Semantics 413

10.4 Natural Language Applications 419
  10.4.1 Story Understanding and Question Answering 419
  10.4.2 A Data Base Front End 420

10.5 Epilogue and References 424

10.6 Exercises 425

11 AUTOMATED REASONING 427
  11.0 Introduction to Weak Methods in Theorem Proving 427
  11.1 The General Problem Solver and Difference Tables 428
  11.2 Resolution Theorem Proving 433
    11.2.1 Introduction 433
    11.2.2 Producing the Clause Form for Resolution Refutations 436
    11.2.3 The Binary Resolution Proof Procedure 440
    11.2.4 Strategies and Simplification Techniques for Resolution 445
    11.2.5 Answer Extraction from Resolution Refutations 449
  11.3 Further Issues in the Design of Automated Reasoning Programs 453
    11.3.1 Uniform Representations for Weak Method Solutions 453
    11.3.2 Alternative Inference Rules 456
    11.3.3 Search Strategies and Their Use 458

11.4 Epilogue and References 459

11.5 Exercises 460

12 MACHINE LEARNING 462
  12.0 Introduction 462
  12.1 A Framework for Learning 465
  12.2 Version Space Search 471
    12.2.1 Generalization Operators and the Concept Space 471
    12.2.2 The Candidate Elimination Algorithm 472
    12.2.3 LEX: Inducing Search Heuristics 479
    12.2.4 Evaluating Candidate Elimination 482
12.3 The ID3 Decision Tree Induction Algorithm 483
12.3.1 Top-Down Decision Tree Induction 486
12.3.2 Information Theoretic Test Selection 487
12.3.3 Evaluating ID3 490
12.4 Inductive Bias and Learnability 490
12.4.1 Inductive Bias 491
12.4.2 The Theory of Learnability 493
12.5 Knowledge and Learning 495
12.5.1 Meta-DENDRAL 496
12.5.2 Explanation-Based Learning 497
12.5.3 EBL and Knowledge Level Learning 501
12.5.4 Analogical Learning 502
12.5.5 Case Based Reasoning 505
12.6 Unsupervised Learning 506
12.6.1 Discovery and Unsupervised Learning 507
12.6.2 Conceptual Clustering 508
12.6.3 COBWEB and the Structure of Taxonomic Knowledge 510
12.7 Parallel Distributed Processing 516
12.7.1 Foundations of Neural Networks 518
12.7.2 The Delta Rule 522
12.7.3 Backpropagation 523
12.7.4 NETtalk 526
12.8 Genetic Algorithms 527
12.8.1 The Genetic Algorithm 528
12.8.2 Evaluating the Genetic Algorithm 529
12.9 Epilogue and References 531
12.10 References 532

PART V
ADVANCED AI PROGRAMMING
TECHNIQUES 535

AI Languages and Meta-Interpreters 535
Object-Oriented Programming 536
Hybrid Environments 537
A Hybrid Example 538
14.3 An Expert System Shell in LISP 610
14.3.1 Implementing Certainty Factors 610
14.3.2 Architecture of lisp-shell 611
14.3.3 User Queries and Working Memory 614
14.3.4 Classification Using lisp-shell 615
14.4 Network Representations and Inheritance 617
14.4.1 Representing Semantic Nets in LISP 618
14.4.2 Implementing Inheritance 620
14.5 The ID3 Induction Algorithm 621
14.5.1 Defining Structures Using defstruct 622
14.5.2 Representing Objects and Properties 624
14.5.3 Data Structures in ID3 626
14.5.4 Implementing ID3 628
14.5.5 Learning Classifications Using build-tree 633
14.5.6 The Object-Oriented Approach to Program Structure 634
14.6 Epilogue and References 635
14.7 Exercises 636

15 Objects, Messages, and Hybrid Expert System Design 638
15.0 Introduction 638
15.1 Object-Oriented Knowledge Representation 639
15.1.1 Objects and Abstraction 639
15.1.2 Benefits of Object-Oriented Programming 646
15.1.3 Object-Oriented Knowledge Bases 647
15.2 LISP and Object-Oriented Programming 649
15.2.1 OOPS: A Simple Object-Oriented Programming Language 649
15.2.2 Implementing OOPS in LISP 652
15.2.3 An Object-Oriented Simulation Using OOPS 656
15.2.4 Evaluating OOPS 660
15.3 The Common LISP Object System 660
15.3.1 Defining Classes and Instances in CLOS 661
15.3.2 Defining Generic Functions and Methods 663
15.3.3 Inheritance in CLOS 665
15.3.4 Advanced Features of CLOS 667
15.3.5 Implementing a Thermostat Simulation 667