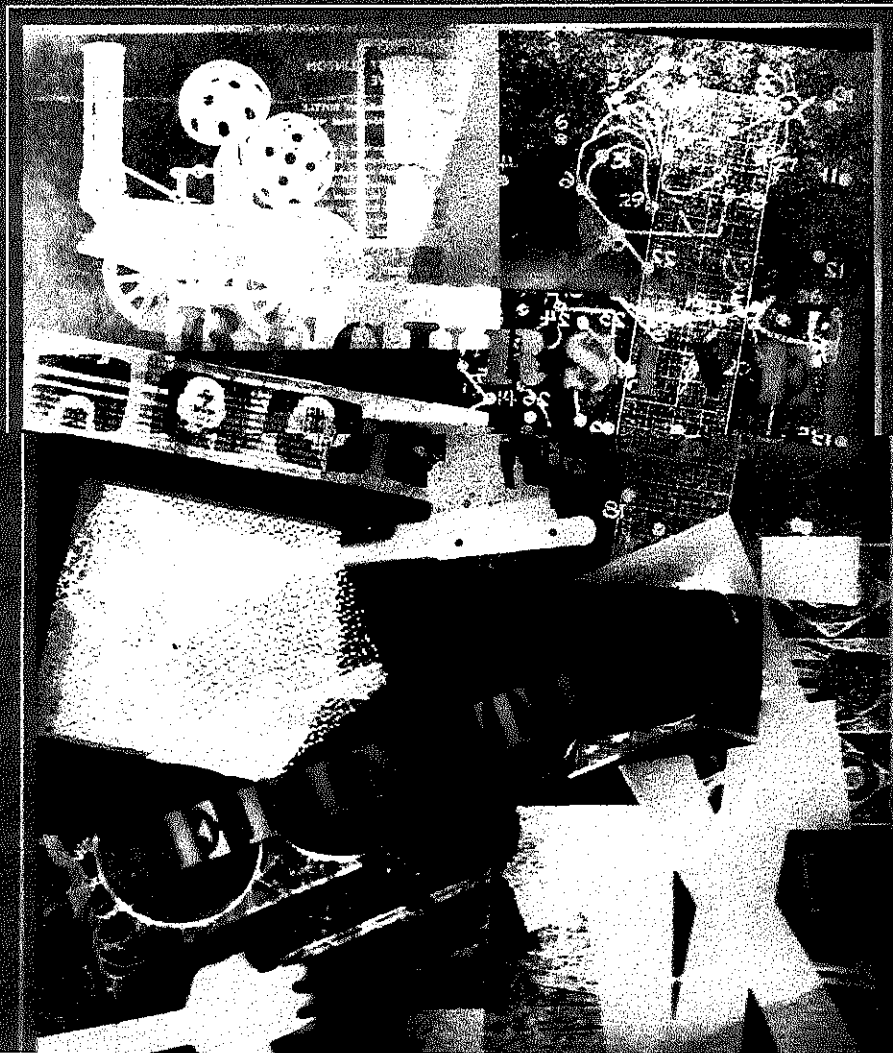


ARTIFICIAL INTELLIGENCE

STRUCTURES AND STRATEGIES FOR COMPLEX PROBLEM SOLVING

SECOND EDITION



1993

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

- 1.3 Artificial Intelligence—A Summary 23
- 1.4 Epilogue and References 24
- 1.5 Exercises 25

PART II

ARTIFICIAL INTELLIGENCE AS REPRESENTATION AND SEARCH 29

- Knowledge Representation 30
- Problem Solving as Search 36

2 THE PREDICATE CALCULUS 41

- 2.0 Introduction 41
- 2.1 The Propositional Calculus 41
 - 2.1.1 Symbols and Sentences 41
 - 2.1.2 The Semantics of the 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 52
- 2.3 Using Inference Rules to Produce Predicate Calculus Expressions 57
 - 2.3.1 Inference Rules 57
 - 2.3.2 Unification 61
 - 2.3.3 A Unification Example 65
- 2.4 Application: A Logic-Based Financial Advisor 67
- 2.5 Epilogue and References 72
- 2.6 Exercises 73

3 STRUCTURES AND STRATEGIES FOR STATE SPACE SEARCH 75

- 3.0 Introduction 75
- 3.1 Graph Theory 78
 - 3.1.1 Structures for State Space Search 78
 - 3.1.2 State Space Representation of Problems 80
- 3.2 Strategies for State Space Search 86
 - 3.2.1 Data-Driven and Goal-Driven Search 86

3.2.2	Implementing Graph Search	89
3.2.3	Depth-First and Breadth-First Search	92
3.2.4	Depth-First Search with Iterative Deepening	99
3.3	Using the State Space to Represent Reasoning with the Predicate Calculus	100
3.3.1	State Space Description of a Logical System	100
3.3.2	And/Or Graphs	101
3.3.3	Further Examples and Applications	104
3.4	Epilogue and References	113
3.5	Exercises	114

4 HEURISTIC SEARCH 116

4.0	Introduction	116
4.1	An Algorithm for Heuristic Search	120
4.1.1	Implementing "Best-First" Search	120
4.1.2	Implementing Heuristic Evaluation Functions	123
4.1.3	Heuristic Search and Expert Systems	130
4.2	Admissibility, Monotonicity, and Informedness	131
4.2.1	Admissibility Measures	132
4.2.2	Monotonicity	134
4.2.3	When One Heuristic Is Better: More Informed Heuristics	135
4.3	Using Heuristics in Games	137
4.3.1	The Minimax Procedure on Exhaustively Searchable Graphs	137
4.3.2	Minimaxing to Fixed Ply Depth	139
4.3.3	The Alpha-Beta Procedure	142
4.4	Complexity Issues	146
4.5	Epilogue and References	148
4.6	Exercises	149

5 CONTROL AND IMPLEMENTATION OF STATE SPACE SEARCH 152

5.0	Introduction	152
5.1	Recursion-Based Search	153
5.1.1	Recursion	153
5.1.2	Recursive Search	154

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

6.3	A Production System Example in PROLOG	228
6.4	Designing Alternative Search Strategies	234
6.4.1	Depth-First Search Using the Closed List	234
6.4.2	Breadth-First Search in PROLOG	235
6.4.3	Best-First Search in PROLOG	237
6.5	A PROLOG Planner	239
6.6	PROLOG: Toward a Nonprocedural Computing Language	242
6.7	Epilogue and References	248
6.8	Exercises	249
7	LISP	252
7.0	Introduction	252
7.1	LISP: A Brief Overview	253
7.1.1	Symbolic Expressions, the Syntactic Basis of LISP	253
7.1.2	Control of LISP Evaluation: quote and eval	257
7.1.3	Programming in LISP: Creating New Functions	258
7.1.4	Program Control in LISP: Conditionals and Predicates	259
7.1.5	Functions, Lists, and Symbolic Computing	262
7.1.6	Lists as Recursive Structures	264
7.1.7	Nested Lists, Structure, and car/cdr Recursion	267
7.1.8	Functional Programming, Side Effects, set, and let	270
7.1.9	Data Types in Common LISP	275
7.2	Search Algorithms in LISP: A Functional Approach to the Farmer, Wolf, Goat, and Cabbage Problem	276
7.3	Higher-Order Functions and Procedural Abstraction	281
7.3.1	Maps and Filters	281
7.3.2	Functional Arguments and Lambda Expressions	284
7.4	Search Strategies in LISP	285
7.4.1	Breadth-First and Depth-First Search	285
7.4.2	Best-First Search	288
7.5	Pattern Matching in LISP	289
7.6	A Recursive Unification Function	291
7.6.1	Implementing the Unification Algorithm	291
7.6.2	Implementing Substitution Sets Using Association Lists	293

- 7.7 Interpreters and Embedded Languages 295
- 7.8 Epilogue and References 298
- 7.9 Exercises 299

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/Shافر, 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 Teiresias 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	NETalk	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

13 **ADVANCED REPRESENTATION IN PROLOG 541**

- 13.0 Introduction 541
- 13.1 PROLOG Tools: Meta-Predicates, Types, and Unification 542
 - 13.1.1 Meta-Logical Predicates 542
 - 13.1.2 Types in PROLOG 543
 - 13.1.3 Unification, the Engine for Predicate Matching and Evaluation 546
- 13.2 Meta-Interpreters in PROLOG 550
 - 13.2.1 An Introduction to Meta-Interpreters: PROLOG in PROLOG 550
 - 13.2.2 Shell for a Rule-Based Expert System 553
 - 13.2.3 Semantic Nets in PROLOG 562
 - 13.2.4 Frames and Schemata in PROLOG 564
- 13.3 Natural Language Understanding in PROLOG 567
 - 13.3.1 Introduction 567
 - 13.3.2 A Recursive Descent Semantic Net Parser 568
- 13.4 Version Space Search in PROLOG 571
 - 13.4.1 The Feature Vector Representation of Concepts and Instances 572
 - 13.4.2 Specific to General Search 573
 - 13.4.3 Candidate Elimination 576
- 13.5 Explanation-Based Learning in PROLOG 579
- 13.6 PROLOG and Programming with Logic 582
 - 13.6.1 Introduction 582
 - 13.6.2 Logic Programming and PROLOG 583
 - 13.6.3 Parallel PROLOG and an Alternative Semantics 588
- 13.7 Epilogue and References 591
- 13.8 Exercises 592

14 **ADVANCED LISP PROGRAMMING TECHNIQUES FOR ARTIFICIAL INTELLIGENCE 595**

- 14.0 Introduction: Abstraction and Complexity 595
- 14.1 Logic Programming in LISP 596
 - 14.1.1 A Simple Logic Programming Language 597
 - 14.1.2 Streams and Stream Processing 599
 - 14.1.3 A Stream-Based Logic Programming Interpreter 601
- 14.2 Streams and Delayed Evaluation 605

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

- 15.4 Object-Oriented Programming and Concurrency in PROLOG 671
 - 15.4.1 Introduction 671
 - 15.4.2 Objects in Concurrent PROLOG 672
- 15.5 Hybrid Expert System Tools 680
 - 15.5.1 Hybrid Environments: Integrating Objects, Rules, and Procedures 680
 - 15.5.2 Facets, Demons, and Graphics in Hybrid Environments 681
- 15.6 Epilogue and References 682
- 15.7 Exercises 683

PART VI EPILOGUE 687

16 ARTIFICIAL INTELLIGENCE AS EMPIRICAL ENQUIRY 687

- 16.0 Introduction 688
- 16.1 Artificial Intelligence: A Revised Definition 689
- 16.2 Cognitive Science: An Overview 694
 - 16.2.1 The Analysis of Human Performance 695
 - 16.2.2 The Production System and Human Cognition 696
- 16.3 Representational Models for Intelligence: Issues and Directions 699
- 16.4 Epilogue and References 703