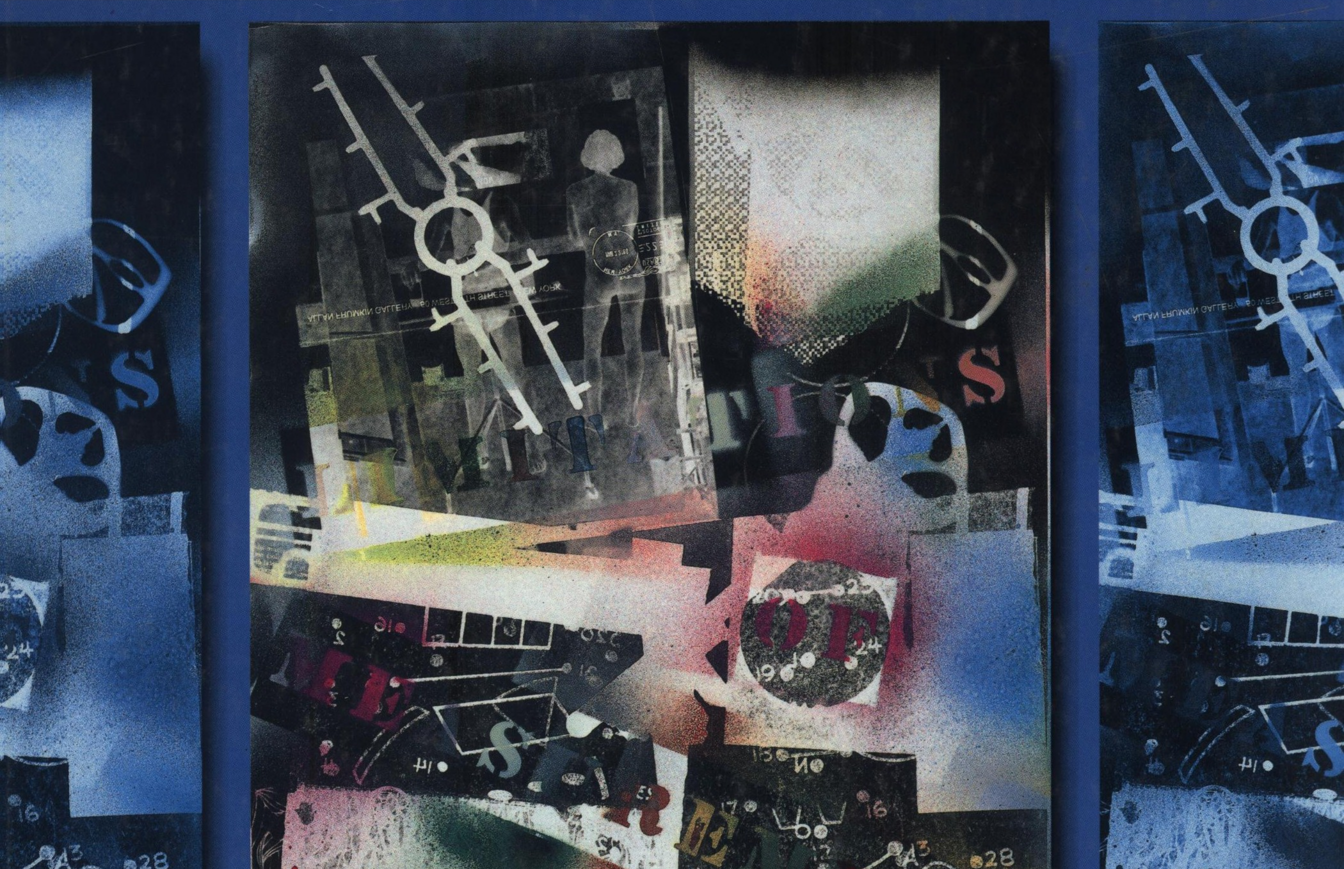
Artificial Intelligence

third edition

Structures and Strategies for Complex Problem Solving



George F Luger & William A Stubblefield

ADDISON-WESLEY

CONTENTS

Preface vii

PART I
ARTIFICIAL INTELLIGENCE: ITS ROOTS AND SCOPE

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

1.1.4 Biological and Social Models of Intelligence: Agent-Oriented Problem Solving 13

1.2 Overview of AI Application Areas 17

1.2.1 Game Playing 18

1.2.2 Automated Reasoning and Theorem Proving 19

1.2.3 Expert Systems 20

1.2.4 Natural Language Understanding and Semantic Modeling 22

1.2.5 Modeling Human Performance 23

1.2.6 Planning and Robotics 23
1.2.7 Languages and Environments for AI 25
1.2.8 Machine Learning 25
1.2.9 Parallel Distributed Processing (PDP) and Emergent Computation 26
1.2.10 AI and Philosophy 27

1.3 Artificial Intelligence—A Summary 28

1.4 Epilogue and References 29

1.5 Exercises 30

PART II
ARTIFICIAL INTELLIGENCE AS REPRESENTATION AND SEARCH

Knowledge Representation 34

Problem Solving as Search 41

2 THE PREDICATE CALCULUS 47

2.0 Introduction 47

2.1 The Propositional Calculus 47

2.1.1 Symbols and Sentences 47
2.1.2 The Semantics of the Propositional Calculus 49

2.2 The Predicate Calculus 52

2.2.1 The Syntax of Predicates and Sentences 52
2.2.2 A Semantics for the Predicate Calculus 58

2.3 Using Inference Rules to Produce Predicate Calculus Expressions 64

2.3.1 Inference Rules 64
2.3.2 Unification 68
2.3.3 A Unification Example 72

2.4 Application: A Logic-Based Financial Advisor 75

2.5 Epilogue and References 79

2.6 Exercises 79

3 STRUCTURES AND STRATEGIES FOR STATE SPACE SEARCH 81

3.0 Introduction 81

3.1 Graph Theory 84

3.1.1 Structures for State Space Search 84

3.1.2 State Space Representation of Problems 87

3.2 Strategies for State Space Search 93

3.2.1 Data-Driven and Goal-Driven Search 93

3.2.2 Implementing Graph Search 96

3.2.3 Depth-First and Breadth-First Search 99

3.2.4 Depth-First Search with Iterative Deepening 106

3.3 Using the State Space to Represent Reasoning with the Predicate Calculus 107

3.3.1 State Space Description of a Logical System 107

3.3.2 And/Or Graphs 109

3.3.3 Further Examples and Applications 111

3.4 Epilogue and References 121

3.5 Exercises 121

4 HEURISTIC SEARCH 123

4.0 Introduction 123

4.1 An Algorithm for Heuristic Search 127

4.1.1 Implementing “Best-First” Search 127

4.1.2 Implementing Heuristic Evaluation Functions 131

4.1.3 Heuristic Search and Expert Systems 136

4.2 Admissibility, Monotonicity, and Informedness 139

4.2.1 Admissibility Measures 139

4.2.2 Monotonicity 141

4.2.3 When One Heuristic Is Better: More Informed Heuristics 142

4.3 Using Heuristics in Games 144

4.3.1 The Minimax Procedure on Exhaustively Searchable Graphs 144

4.3.2 Minimaxing to Fixed Ply Depth 147

4.3.3 The Alpha-Beta Procedure 150

4.4 Complexity Issues 152

4.5 Epilogue and References 156

4.6 Exercises 156

5 CONTROL AND IMPLEMENTATION OF STATE SPACE SEARCH 159

5.0 Introduction 159

5.1 Recursion-Based Search 160

5.1.1 Recursion 160

5.1.2 Recursive Search 161

5.2 Pattern-Directed Search 164

5.2.1 Example: Recursive Search in the Knight's Tour Problem 165

5.2.2 Refining the Pattern-search Algorithm 168

5.3 Production Systems 171

5.3.1 Definition and History 171

5.3.2 Examples of Production Systems 174

5.3.3 Control of Search in Production Systems 180

5.3.4 Advantages of Production Systems for AI 184

5.4 Predicate Calculus and Planning 186

5.5 The Blackboard Architecture for Problem Solving 196

5.6 Epilogue and References 198

5.7 Exercises 199

PART III
REPRESENTATIONS FOR KNOWLEDGE-BASED PROBLEM SOLVING

6 KNOWLEDGE-INTENSIVE PROBLEM SOLVING 207

6.0 Introduction 207

6.1 Overview of Expert System Technology 210

6.1.1 The Design of Rule-Based Expert Systems 210
6.1.2 Selecting a Problem for Expert System Development 212
6.1.3 The Knowledge Engineering Process 214
6.1.4 Conceptual Models and Their Role in Knowledge Acquisition 216

6.2 Rule-based Expert Systems 219

6.2.1 The Production System and Goal-driven Problem Solving 220
6.2.2 Explanation and Transparency in Goal-driven Reasoning 224
6.2.3 Using the Production System for Data-driven Reasoning 226
6.2.4 Heuristics and Control in Expert Systems 229
6.2.5 Conclusions: Rule-Based Reasoning 230

6.3 Model-based Reasoning 231

6.3.1 Introduction 231

6.4 Case-based Reasoning 235

6.4.1 Introduction 235

6.5 The Knowledge-Representation Problem 240

6.6 Epilogue and References 245

6.7 Exercises 246

7 REASONING WITH UNCERTAIN OR INCOMPLETE INFORMATION 247

7.0 Introduction 247

7.1 The Statistical Approach to Uncertainty 249

7.1.1 Bayesian Reasoning 250
7.1.2 Bayesian Belief Networks 254
7.1.3 The Dempster–Shafer Theory of Evidence 259
7.1.4 The Stanford Certainty Factor Algebra 263
7.1.5 Causal Networks 266

7.2 Introduction to Nonmonotonic Systems 269

7.2.1 Logics for Nonmonotonic Reasoning 269
7.2.2 Logics Based on Minimum Models 273
7.2.3 Truth Maintenance Systems 275
7.2.4 Set Cover and Logic Based Abduction (Stern 1996) 281

7.3 Reasoning with Fuzzy Sets 284

7.4 Epilogue and References 289

7.5 Exercises 290

8 KNOWLEDGE REPRESENTATION 293

8.0 Knowledge Representation Languages 293

8.1 Issues in Knowledge Representation 295

8.2 A Survey of Network Representation 297

8.2.1 Associationist Theories of Meaning 297
8.2.2 Early Work in Semantic Nets 301
8.2.3 Standardization of Network Relationships 303

8.3 Conceptual Graphs: A Network Representation Language 309

8.3.1 Introduction to Conceptual Graphs 309

8.3.2 Types, Individuals, and Names 311

8.3.3 The Type Hierarchy 313

8.3.4 Generalization and Specialization 314

8.3.5 Propositional Nodes 317

8.3.6 Conceptual Graphs and Logic 318

8.4 Structured Representations 320

8.4.1 Frames 320

8.4.2 Scripts 324

8.5 Issues in Knowledge Representation 328

8.5.1 Hierarchies, Inheritance, and Exceptions 328

8.5.2 Naturalness, Efficiency, and Plasticity 331

8.6 Epilogue and References 334

8.7 Exercises 335

PART IV
LANGUAGES AND PROGRAMMING TECHNIQUES FOR ARTIFICIAL INTELLIGENCE

Languages, Understanding, and Levels of Abstraction 340

Desired Features of AI Language 342

An Overview of LISP and PROLOG 349

Object-Oriented Programming 352

Hybrid Environments 353

A Hybrid Example 354

Selecting an Implementation Language 356

9 AN INTRODUCTION TO PROLOG 357

9.0 Introduction 357

9.1 Syntax for Predicate Calculus Programming 358

9.1.1 Representing Facts and Rules 358
9.1.2 Creating, Changing, and Monitoring the PROLOG Environment 362
9.1.3 Recursion-Based Search in PROLOG 364
9.1.4 Recursive Search in PROLOG 366
9.1.5 The Use of Cut to Control Search in PROLOG 369

9.2 Abstract Data Types (ADTs) in PROLOG 371

9.2.1 The ADT Stack 371
9.2.2 The ADT Queue 373
9.2.3 The ADT Priority Queue 373
9.2.4 The ADT Set 374

9.3 A Production System Example in PROLOG 375

9.4 Designing Alternative Search Strategies 381

9.4.1 Depth-First Search Using the Closed List 381
9.4.2 Breadth-First Search in PROLOG 383
9.4.3 Best-First Search in PROLOG 384

9.5 A PROLOG Planner 386

9.6 PROLOG: Meta-Predicates, Types, and Unification 389

9.6.1 Meta-Logical Predicates 389
9.6.2 Types in PROLOG 391
9.6.3 Unification, the Engine for Predicate Matching and Evaluation 394

9.7 Meta-Interpreters in PROLOG 397

9.7.1 An Introduction to Meta-Interpreters: PROLOG in PROLOG 397
9.7.2 Shell for a Rule-Based Expert System 401
9.7.3 Semantic Nets in PROLOG 410
9.7.4 Frames and Schemata in PROLOG 412

9.8 PROLOG: Towards Nonprocedural Computing 415

9.9 Epilogue and References 421

9.10 Exercises 422

10 AN INTRODUCTION TO LISP 425

10.0 Introduction 425

10.1 LISP: A Brief Overview 426

10.1.1 Symbolic Expressions, the Syntactic Basis of LISP 426
10.1.2 Control of LISP Evaluation: quote and eval 430
10.1.3 Programming in LISP: Creating New Functions 431
10.1.4 Program Control in LISP: Conditionals and Predicates 433
10.1.5 Functions, Lists, and Symbolic Computing 436
10.1.6 Lists as Recursive Structures 438
10.1.7 Nested Lists, Structure, and car/cdr Recursion 441
10.1.8 Binding Variables Using set 444
10.1.9 Defining Local Variables Using let 446
10.1.10 Data Types in Common LISP 448
10.1.11 Conclusion 449

10.2 Search in LISP: A Functional Approach to the Farmer, Wolf, Goat, and Cabbage Problem 449

10.3 Higher-Order Functions and Procedural Abstraction 455

10.3.1 Maps and Filters 455
10.3.2 Functional Arguments and Lambda Expressions 457

10.4 Search Strategies in LISP 459

10.4.1 Breadth-First and Depth-First Search 459
10.4.2 Best-First Search 462

10.5 Pattern Matching in LISP 463

10.6 A Recursive Unification Function 465

10.6.1 Implementing the Unification Algorithm 465
10.6.2 Implementing Substitution Sets Using Association Lists 467

10.7 Interpreters and Embedded Languages 469

10.8 Logic Programming in LISP 472

10.8.1 A Simple Logic Programming Language 472
10.8.2 Streams and Stream Processing 474
10.8.3 A Stream-Based Logic Programming Interpreter 477

10.9 Streams and Delayed Evaluation 482

10.10 An Expert System Shell in LISP 486

10.10.1 Implementing Certainty Factors 486
10.10.2 Architecture of lisp-shell 488
10.10.3 User Queries and Working Memory 490
10.10.4 Classification Using lisp-shell 491

10.11 Network Representations and Inheritance 494

10.11.1 Representing Semantic Nets in LISP 494
10.11.2 Implementing Inheritance 497

10.12 Object-Oriented Programming Using CLOS 497

10.12.1 Defining Classes and Instances in CLOS 499
10.12.2 Defining Generic Functions and Methods 501
10.12.3 Inheritance in CLOS 503
10.12.4 Advanced Features of CLOS 505
10.12.5 Example: A Thermostat Simulation 505

10.13 Epilogue and References 511

10.14 Exercises 511

PART V
ADVANCED TOPICS FOR AI PROBLEM SOLVING

Natural Language, Automated Reasoning, and Learning 517

11 UNDERSTANDING NATURAL LANGUAGE 519

11.0 Role of Knowledge in Language Understanding 519

11.1 Language Understanding: A Symbolic Approach 522

11.1.1 Introduction 522
11.1.2 Stages of Language Analysis 523

11.2 Syntax 524

11.2.1 Specification and Parsing Using Context-Free Grammars 524
11.2.2 Transition Network Parsers 527
11.2.3 The Chomsky Hierarchy and Context-Sensitive Grammars 531

11.3 Combining Syntax and Semantics in ATN Parsers 534

11.3.1 Augmented Transition Network Parsers 534
11.3.2 Combining Syntax and Semantics 538

11.4 Stochastic Tools for Language Analysis 543

11.4.1 Introduction 543
11.4.2 A Markov Model Approach 545
11.4.3 A CART Tree Approach 546
11.4.4 Mutual Information Clustering 547
11.4.5 Parsing 548
11.4.6 Other Language Applications for Stochastic Techniques 550

11.5 Natural Language Applications 550

11.5.1 Story Understanding and Question Answering 550
11.5.2 A Database Front End 551

11.6 Epilogue and References 555

11.7 Exercises 557

12 AUTOMATED REASONING 559

12.0 Introduction to Weak Methods in Theorem Proving 559

12.1 The General Problem Solver and Difference Tables 560

12.2 Resolution Theorem Proving 566

12.2.1 Introduction 566

12.2.2 Producing the Clause Form for Resolution Refutations 568

12.2.3 The Binary Resolution Proof Procedure 573

12.2.4 Strategies and Simplification Techniques for Resolution 578

12.2.5 Answer Extraction from Resolution Refutations 583

12.3 PROLOG and Automated Reasoning 587

12.3.1 Introduction 587

12.3.2 Logic Programming and PROLOG 588

12.4 Further Issues in Automated Reasoning 593

12.4.1 Uniform Representations for Weak Method Solutions 593

12.4.2 Alternative Inference Rules 597

12.4.3 Search Strategies and Their Use 599

12.5 Epilogue and References 600

12.6 Exercises 601

13 MACHINE LEARNING: SYMBOL-BASED 603

13.0 Introduction 603

13.1 A Framework for Symbol-based Learning 606

13.2 Version Space Search 612

13.2.1 Generalization Operators and the Concept Space 612

13.2.2 The Candidate Elimination Algorithm 613

13.2.3 LEX: Inducing Search Heuristics 620

13.2.4 Evaluating Candidate Elimination 623

13.3 The ID3 Decision Tree Induction Algorithm 624

13.3.1 Top-Down Decision Tree Induction 627
13.3.2 Information Theoretic Test Selection 628
13.3.3 Evaluating ID3 632
13.3.4 Decision Tree Data Issues: Bagging, Boosting 632

13.4 Inductive Bias and Learnability 633

13.4.1 Inductive Bias 634
13.4.2 The Theory of Learnability 636

13.5 Knowledge and Learning 638

13.5.1 Meta-DENDRAL 639
13.5.2 Explanation-Based Learning 640
13.5.3 EBL and Knowledge-Level Learning 645
13.5.4 Analogical Reasoning 646

13.6 Unsupervised Learning 649

13.6.1 Discovery and Unsupervised Learning 649
13.6.2 Conceptual Clustering 651
13.6.3 COBWEB and the Structure of Taxonomic Knowledge 653

13.7 Epilogue and References 658

13.8 Exercises 659

14 MACHINE LEARNING: CONNECTIONIST 661

14.0 Introduction 661

14.1 Foundations for Connectionist Networks 663

14.1.1 Early History 663

14.2 Perceptron Learning 666

14.2.1 The Perceptron Training Algorithm 666
14.2.2 An Example: Using a Perceptron Network to Classify 668
14.2.3 The Delta Rule 672

14.3 Backpropagation Learning 675

14.3.1 Deriving the Backpropagation Algorithm 675
14.3.2 Backpropagation Example 1: NETtalk 679
14.3.3 Backpropagation Example 2: Exclusive-or 681

14.4 Competitive Learning 682

14.4.1 Winner-Take-All Learning for Classification 682
14.4.2 A Kohonen Network for Learning Prototypes 684
14.4.3 Grossberg Learning and Counterpropagation 686

14.5 Hebbian Coincidence Learning 690

14.5.1 Introduction 690
14.5.2 An Example of Unsupervised Hebbian Learning 691
14.5.3 Supervised Hebbian Learning 694
14.5.4 Associative Memory and the Linear Associator 696

14.6 Attractor Networks or "Memories" 701

14.6.1 Introduction 701
14.6.2 BAM, the Bi-directional Associative Memory 702
14.6.3 Examples of BAM Processing 704
14.6.4 Autoassociative Memory and Hopfield Nets 706

14.7 Epilogue and References 711

14.8 Exercises 712

15 MACHINE LEARNING: SOCIAL AND EMERGENT 713

15.0 Social and Emergent Models of Learning 713

15.1 The Genetic Algorithm 715

15.1.3 Two Examples: CNF Satisfaction and the Traveling Salesperson 717
15.1.4 Evaluating the Genetic Algorithm 721

15.2 Classifier Systems and Genetic Programming 725

15.2.1 Classifier Systems 725
15.2.2 Programming with Genetic Operators 730

15.3 Artificial Life and Society-based Learning 736

- 15.3.1 The "Game of Life" 737
- 15.3.2 Evolutionary Programming 740
- 15.3.3 A Case Study in Emergence (Crutchfield and Mitchell 1994) 743

15.4 Epilogue and References 747

15.5 Exercises 748

PART VI
EPILOGUE

Reflections on the Nature of Intelligence 751

16 ARTIFICIAL INTELLIGENCE AS EMPIRICAL ENQUIRY 753

16.0 Introduction 753

16.1 Artificial Intelligence: A Revised Definition 755

- 16.1.1 Intelligence and the Physical Symbol System 756
- 16.1.2 Minds, Brains, and Neural Computing 759
- 16.1.3 Agents, Emergence, and Intelligence 761
- 16.1.4 Situated Actors and the Existential Mind 764

16.2 Cognitive Science: An Overview 766

- 16.2.1 The Analysis of Human Performance 766
- 16.2.2 The Production System and Human Cognition 767

16.3 Current Issues in Machine Learning 770

16.4 Understanding Intelligence: Issues and Directions 775

16.5 Epilogue and References 780

Bibliography 781
Author Index 803
Subject Index 809
Acknowledgements 823