ARTIFICIAL INTELLIGENCE

SECOND EDITION

ELAINE RICH
KEVIN KNIGHT

TATA McGRAW-HILL EDITION
FOR SALE IN INDIA ONLY

Artificial Intelligence

Second Edition

Elaine Rich

Microelectronics and Computer Technology Corporation

Kevin Knight

Carnegie Mellon University

Tata McGraw-Hill Publishing Company Limited
NEW DELHI

McGraw-Hill Offices

New Delhi New York St Louis San Francisco Auckland
Bogota' Guatemala Hamburg Lisbon London Madrid
Mexico Milan Montreal Panama Paris San Juan São Paulo
Singapore Sydney Tokyo Toronto

Tata McGraw-Hill
*A Division of The **McGraw-Hill** Companies*

ARTIFICIAL INTELLIGENCE

Tata McGraw-Hill Edition 1991

24th reprint 2004
RALLRRYLRZZLD

Reprinted in India by arrangement with The McGraw-Hill Companies, Inc., New York

ISBN 0-07-460081-8

Published by Tata McGraw-Hill Publishing Company Limited,
7 West Patel Nagar, New Delhi 110 008, and printed at
Pushp Print Services, Delhi 110 053

About the Authors

Elaine Rich is Director of the Artificial Intelligence Laboratory at the Microelectronics and Computer Technology Corporation (MCC). Formerly on the faculty in computer sciences at the University of Texas, she received her Ph.D. from Carnegie Mellon University. Her research interests include natural language processing, knowledge representation, and machine translation.

Kevin Knight received his B.A. from Harvard University and is presently completing his Ph.D. in computer science at Carnegie Mellon University. A regular consultant at MCC, his research interests include natural language processing, unification, and machine translation, and search.

For our fathers

Robert Rich

and

Gary Knight

Contents

Preface xv

I Problems and Search 1

1 What Is Artificial Intelligence? 3
1.1 The AI Problems 3
1.2 The Underlying Assumption 6
1.3 What Is an AI Technique? 8
1.4 The Level of the Model 22
1.5 Criteria for Success 24
1.6 Some General References 26
1.7 One Final Word 27
1.8 Exercises 28

2 Problems, Problem Spaces, and Search 29
2.1 Defining the Problem as a State Space Search 29
2.2 Production Systems 36
2.3 Problem Characteristics 44
2.4 Production System Characteristics 55
2.5 Issues in the Design of Search Programs 57
2.6 Additional Problems 60
2.7 Summary 61
2.8 Exercises 61

3 Heuristic Search Techniques 63
3.1 Generate-and-Test 64
3.2 Hill Climbing 65
3.3 Best-First Search 73
3.4 Problem Reduction 82
3.5 Constraint Satisfaction 88
3.6 Means-Ends Analysis 94
3.7 Summary 97
3.8 Exercises 98

II Knowledge Representation 103

4 Knowledge Representation Issues 105
4.1 Representations and Mappings 105
4.2 Approaches to Knowledge Representation 109
4.3 Issues in Knowledge Representation 115
4.4 The Frame Problem 126
4.5 Summary 129

5 Using Predicate Logic 131
5.1 Representing Simple Facts in Logic 131
5.2 Representing Instance and Isa Relationships 137
5.3 Computable Functions and Predicates 139
5.4 Resolution 143
5.5 Natural Deduction 164
5.6 Summary 165
5.7 Exercises 166

6 Representing Knowledge Using Rules 171
6.1 Procedural versus Declarative Knowledge 171
6.2 Logic Programming 173
6.3 Forward versus Backward Reasoning 177
6.4 Matching 182
6.5 Control Knowledge 188
6.6 Summary 192
6.7 Exercises 192

7 Symbolic Reasoning under Uncertainty 195
7.1 Introduction to Nonmonotonic Reasoning 195
7.2 Logics for Nonmonotonic Reasoning 199
7.3 Implementation Issues 208
7.4 Augmenting a Problem Solver 209
7.5 Implementation: Depth-First Search 211
7.6 Implementation: Breadth-First Search 222
7.7 Summary 226
7.8 Exercises 227

8 Statistical Reasoning 231
8.1 Probability and Bayes' Theorem 231
8.2 Certainty Factors and Rule-Based Systems 233
8.3 Bayesian Networks 239
8.4 Dempster-Shafer Theory 242
8.5 Fuzzy Logic 246
8.6 Summary 247
8.7 Exercises 248

9 Weak Slot-and-Filler Structures 251
9.1 Semantic Nets 251
9.2 Frames 257
9.3 Exercises 275

10 Strong Slot-and-Filler Structures 277
10.1 Conceptual Dependency 277
10.2 Scripts 284
10.3 CYC 288
10.4 Exercises 294

11 Knowledge Representation Summary 297
11.1 Syntactic-Semantic Spectrum of Representation 297
11.2 Logic and Slot-and-Filler Structures 299
11.3 Other Representational Techniques 301
11.4 Summary of the Role of Knowledge 302
11.5 Exercises 303

III Advanced Topics 305

12 Game Playing 307
12.1 Overview 307
12.2 The Minimax Search Procedure 310
12.3 Adding Alpha-Beta Cutoffs 314
12.4 Additional Refinements 319
12.5 Iterative Deepening 322
12.6 References on Specific Games 324
12.7 Exercises 326

13 Planning 329
13.1 Overview 329
13.2 An Example Domain: The Blocks World 332
13.3 Components of a Planning System 333
13.4 Goal Stack Planning 339
13.5 Nonlinear Planning Using Constraint Posting 347
13.6 Hierarchical Planning 354
13.7 Reactive Systems 356
13.8 Other Planning Techniques 357
13.9 Exercises 357

14 Understanding 359
14.1 What Is Understanding? 359
14.2 What Makes Understanding Hard? 360
14.3 Understanding as Constraint Satisfaction 367
14.4 Summary 375
14.5 Exercises 375

15 Natural Language Processing 377
15.1 Introduction 379
15.2 Syntactic Processing 385
15.3 Semantic Analysis 397
15.4 Discourse and Pragmatic Processing 415
15.5 Summary 424
15.6 Exercises 426

16 Parallel and Distributed AI 429
16.1 Psychological Modeling 429
16.2 Parallelism in Reasoning Systems 430
16.3 Distributed Reasoning Systems 433
16.4 Summary 445
16.5 Exercises 445

17 Learning 447
17.1 What Is Learning? 447
17.2 Rote Learning 448
17.3 Learning by Taking Advice 450
17.4 Learning in Problem Solving 452
17.5 Learning from Examples: Induction 457
17.6 Explanation-Based Learning 471
17.7 Discovery 475
17.8 Analogy 479
17.9 Formal Learning Theory 482
17.10 Neural Net Learning and Genetic Learning 483
17.11 Summary 483
17.12 Exercises 484

18 Connectionist Models 487
18.1 Introduction: Hopfield Networks 488
18.2 Learning in Neural Networks 492
18.3 Applications of Neural Networks 514
18.4 Recurrent Networks 517
18.5 Distributed Representations 520
18.6 Connectionist AI and Symbolic AI 522
18.7 Exercises 525

19 Common Sense 529
19.1 Qualitative Physics 530
19.2 Commonsense Ontologies 533
19.3 Memory Organization 540
19.4 Case-Based Reasoning 543
19.5 Exercises 545

20 Expert Systems 547
20.1 Representing and Using Domain Knowledge 547
20.2 Expert System Shells 549
20.3 Explanation 550
20.4 Knowledge Acquisition 553
20.5 Summary 556
20.6 Exercises 557

21 Perception and Action 559
21.1 Real-Time Search 561
21.2 Perception 563
21.3 Action 569
21.4 Robot Architectures 573
21.5 Summary 576
21.6 Exercises 577

22 Conclusion 579
22.1 Components of an AI Program 579
22.2 Exercises 580

References 583

Acknowledgements 605

Author Index 607

Subject Index 613

Preface

In the years since the first edition of this book appeared, Artificial Intelligence (AI) has grown from a small-scale laboratory science into a technological and industrial success. We now possess an arsenal of techniques for creating computer programs that control manufacturing processes, diagnose computer faults and human diseases, design computers, do insurance underwriting, play grandmaster-level chess, and so on. Basic research in AI has expanded enormously during this period. For the student, extracting theoretical and practical knowledge from such a large body of scientific knowledge is a daunting task. The goal of the first edition of this book was to provide a readable introduction to the problems and techniques of AI. In this edition, we have tried to achieve the same goal for the expanded field that AI has become. In particular, we have tried to present both the theoretical foundations of AI and an indication of the ways that current techniques can be used in application programs.

As a result of this effort, the book has grown. It is probably no longer possible to cover everything in a single semester. Because of this, we have structured the book so that an instructor can choose from a variety of paths through the chapters. The book is divided into three parts:

Part I. Problems and Search
Part II. Knowledge Representation.
Part III. Advanced Topics.

Part I introduces AI by examining the nature of the difficult problems that AI seeks to solve. It then develops the theory and practice of heuristic search, providing detailed algorithms for standard search methods, including best-first search, hill climbing, simulated annealing, means-ends analysis, and constraint satisfaction.

The last thirty years of AI have demonstrated that intelligence requires more than the ability to reason. It also requires a great deal of knowledge about the world. So Part II explores a variety of methods for encoding knowledge in computer systems. These methods include predicate logic, production rules, semantic networks, frames, and scripts. There are also chapters on both symbolic and numeric techniques for reasoning under uncertainty. In addition, we present some very specific frameworks in which particular commitments to a set of representational primitives are made.

Parts I and II should be covered in any basic course in AI. They provide the foundation for the advanced topics and applications that are presented in Part III. While the chapters in Parts I and II should be covered in order since they build on each other, the chapters in Part III are, for the most part, independent and can be covered in almost any combination, depending on the goals of a particular course. The topics that are covered include: game playing, planning, understanding, natural language processing (which depends on the understanding chapter), parallel and distributed AI (which depends on planning and natural language), learning, connectionist models, common sense, expert systems, and perception and action.

To use this book effectively, students should have some background in both computer science and mathematics. As computer science background, they should have experience programming and they should feel comfortable with the material in an undergraduate data structures course. They should be familiar with the use of recursion as a program control structure. And they should be able to do simple analyses of the time complexity of algorithms. As mathematical background, students should have the equivalent of an undergraduate course in logic, including predicate logic with quantifiers and the basic notion of a decision procedure.

This book contains, spread throughout it, many references to the AI research literature. These references are important for two reasons. First, they make it possible for the student to pursue individual topics in greater depth than is possible within the space restrictions of this book. This is the common reason for including references in a survey text. The second reason that these references have been included is more specific to the content of this book. AI is a relatively new discipline. In many areas of the field there is still not complete agreement on how things should be done. The references to the source literature guarantee that students have access not just to one approach, but to as many as possible of those whose eventual success still needs to be determined by further research, both theoretical and empirical.

Since the ultimate goal of AI is the construction of programs that solve hard problems, no study of AI is complete without some experience writing programs. Most AI programs are written in LISP, PROLOG, or some specialized AI shell. Recently though, as AI has spread out into the mainstream computing world, AI programs are being written in a wide variety of programming languages. The algorithms presented in this book are described in sufficient detail to enable students to exploit them in their programs, but they are not expressed in code. This book should probably be supplemented with a good book on whatever language is being used for programming in the course.

This book would not have happened without the help of many people. The content of the manuscript has been greatly improved by the comments of Srinivas Akella, Jim Blevins, Clay Bridges, R. Martin Chavez, Alan Cline, Adam Farquar, Anwar Ghuloum, Yolanda Gil, R. V. Guha, Lucy Hadden, Ajay Jain, Craig Knoblock, John Laird, Clifford Mercer, Michael Newton, Charles Petrie, Robert Rich, Steve Shafer, Reid Simmons, Herbert Simon, Munindar Singh, Milind Tambe, David Touretzky, Manuela Veloso, David Wroblewski, and Marco Zagha.

Special thanks to Yolanda Gil and Alan Cline for help above and beyond. Yolanda kept the project going under desperate circumstances, and Alan spent innumerable hours designing the cover and bringing it into the world. We thank them for these things and much, much more.

Linda Mitchell helped us put together many draft editions along the way. Some of those drafts were used in actual courses, where students found innumerable bugs for us. We would like to thank them as well as their instructors, Tom Mitchell and Jean Scholtz. Thanks also to Don Speray for his help in producing the cover.

David Shapiro and Joe Murphy deserve credit for superb editing, and for keeping us on schedule.

We would also like to thank Nicole Vecchi for her wisdom and patience in the world of high resolution printing. Thanks to David Long and Lily Mummert for pointing us to the right fonts.

Thanks to the following reviewers for their comments: Yigal Arens, University of Southern California; Jaime Carbonell, Carnegie Mellon University; Charles Dyer, University of Wisconsin, Madison; George Ernst, Case Western Reserve University; Pat Langley, University of California, Irvine; Brian Schunck, University of Michigan; and James Slagle, University of Minnesota.

Carnegie Mellon University and MCC provided us the environment in which we could write and produce this book. We would like to thank our colleagues, particularly Jim Barnett and Masaru Tomita, for putting up with us while we were writing this book instead of doing the other things we were supposed to be doing.

Elaine Rich
Kevin Knight