

# Artificial Intelligence

## Foundations of Computational Agents

Artificial intelligence, including machine learning, has emerged as a transformational science and engineering discipline. *Artificial Intelligence: Foundations of Computational Agents* presents AI using a coherent framework to study the design of intelligent computational agents. By showing how the basic approaches fit into a multidimensional design space, readers learn the fundamentals without losing sight of the bigger picture. The new edition also features expanded coverage on machine learning material, as well as on the social and ethical consequences of AI and ML. The book balances theory and experiment, showing how to link them together, and develops the science of AI together with its engineering applications. Although structured as an undergraduate and graduate textbook, the book's straightforward, self-contained style will also appeal to an audience of professionals, researchers, and independent learners. The second edition is well-supported by strong pedagogical features and online resources to enhance student comprehension.

David Poole is a Professor of Computer Science at the University of British Columbia. He is a co-author of three artificial intelligence books including *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. He is a former Chair of the Association for Uncertainty in Artificial Intelligence, the winner of the Canadian AI Association (CAIAC) 2013 Lifetime Achievement Award, and a Fellow of the Association for the Advancement Artificial Intelligence (AAAI) and CAIAC.

Alan Mackworth is a Professor of Computer Science at the University of British Columbia. He has authored over 130 papers and co-authored two books: *Computational Intelligence: A Logical Approach* and *Artificial Intelligence: Foundations of Computational Agents*. His awards include the AIJ Classic Paper Award and the ACP Award for Research Excellence. He has served as President of IJCAI, AAAI and CAIAC. He is a Fellow of AAAI, CAIAC and the Royal Society of Canada.

# Artificial Intelligence

## Foundations of Computational Agents

David L. Poole

*University of British Columbia, Canada*

Alan K. Mackworth

*University of British Columbia, Canada*



CAMBRIDGE  
UNIVERSITY PRESS

**CAMBRIDGE**  
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom  
One Liberty Plaza, 20th Floor, New York, NY 10006, USA  
477 Williamstown Road, Port Melbourne, VIC 3207, Australia  
4843/24, 2nd Floor, Ansari Road, Daryaganj, Delhi - 110002, India  
79 Anson Road, #06-04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.  
It furthers the University's mission by disseminating knowledge in the pursuit of  
education, learning and research at the highest international levels of excellence.

[www.cambridge.org](http://www.cambridge.org)  
Information on this title: [www.cambridge.org/9781107195394](http://www.cambridge.org/9781107195394)  
DOI: 10.1017/9781108164085

© David L. Poole and Alan K. Mackworth

This publication is in copyright. Subject to statutory exception  
and to the provisions of relevant collective licensing agreements,  
no reproduction of any part may take place without the written  
permission of Cambridge University Press.

First edition published 2010  
Second edition published 2017

Printed in United States of America by Sheridan Books, Inc

*A catalogue record for this publication is available from the British Library*

ISBN 978-1-107-19539-4 Hardback

Cambridge University Press has no responsibility for the persistence or accuracy  
of URLs for external or third-party internet websites referred to in this publication,  
and does not guarantee that any content on such websites is, or will remain,  
accurate or appropriate.

# Contents

<b>Contents</b>	<b>v</b>
<b>Figures</b>	<b>xv</b>
<b>I Agents in the World: What are Agents and How Can They be Built?</b>	<b>1</b>
<b>1 Artificial Intelligence and Agents</b>	<b>3</b>
1.1 What is Artificial Intelligence? . . . . .	3
1.1.1 Artificial and Natural Intelligence . . . . .	5
1.2 A Brief History of Artificial Intelligence . . . . .	7
1.2.1 Relationship to Other Disciplines . . . . .	10
1.3 Agents Situated in Environments . . . . .	11
1.4 Designing Agents . . . . .	13
1.4.1 Design Time, Offline and Online Computation . . . . .	13
1.4.2 Tasks . . . . .	15
1.4.3 Defining a Solution . . . . .	17
1.4.4 Representations . . . . .	19
1.5 Agent Design Space . . . . .	21
1.5.1 Modularity . . . . .	21
1.5.2 Planning Horizon . . . . .	23
1.5.3 Representation . . . . .	23
1.5.4 Computational Limits . . . . .	25
1.5.5 Learning . . . . .	27

1.5.6	Uncertainty . . . . .	28
1.5.7	Preference . . . . .	29
1.5.8	Number of Agents . . . . .	30
1.5.9	Interaction . . . . .	30
1.5.10	Interaction of the Dimensions . . . . .	31
1.6	Prototypical Applications . . . . .	33
1.6.1	An Autonomous Delivery Robot . . . . .	34
1.6.2	A Diagnostic Assistant . . . . .	36
1.6.3	An Intelligent Tutoring System . . . . .	39
1.6.4	A Trading Agent . . . . .	41
1.6.5	Smart House . . . . .	43
1.7	Overview of the Book . . . . .	44
1.8	Review . . . . .	45
1.9	References and Further Reading . . . . .	46
1.10	Exercises . . . . .	47
<b>2</b>	<b>Agent Architectures and Hierarchical Control</b>	<b>49</b>
2.1	Agents . . . . .	50
2.2	Agent Systems . . . . .	51
2.2.1	The Agent Function . . . . .	51
2.3	Hierarchical Control . . . . .	56
2.4	Acting with Reasoning . . . . .	65
2.4.1	Agents Modeling the World . . . . .	65
2.4.2	Knowledge and Acting . . . . .	66
2.4.3	Design Time and Offline Computation . . . . .	67
2.4.4	Online Computation . . . . .	69
2.5	Review . . . . .	70
2.6	References and Further Reading . . . . .	71
2.7	Exercises . . . . .	71
<b>II</b>	<b>Reasoning, Planning and Learning with Certainty</b>	<b>75</b>
<b>3</b>	<b>Searching for Solutions</b>	<b>77</b>
3.1	Problem Solving as Search . . . . .	77
3.2	State Spaces . . . . .	79
3.3	Graph Searching . . . . .	81
3.3.1	Formalizing Graph Searching . . . . .	82
3.4	A Generic Searching Algorithm . . . . .	85
3.5	Uninformed Search Strategies . . . . .	87
3.5.1	Breadth-First Search . . . . .	87
3.5.2	Depth-First Search . . . . .	90
3.5.3	Iterative Deepening . . . . .	94

Contents	vii
3.5.4	Lowest-Cost-First Search . . . . . 97
3.6	Heuristic Search . . . . . 98
3.6.1	A* Search . . . . . 100
3.6.2	Designing a Heuristic Function . . . . . 104
3.7	Pruning the Search Space . . . . . 105
3.7.1	Cycle Pruning . . . . . 105
3.7.2	Multiple-Path Pruning . . . . . 106
3.7.3	Summary of Search Strategies . . . . . 109
3.8	More Sophisticated Search . . . . . 110
3.8.1	Branch and Bound . . . . . 110
3.8.2	Direction of Search . . . . . 113
3.8.3	Dynamic Programming . . . . . 115
3.9	Review . . . . . 119
3.10	References and Further Reading . . . . . 119
3.11	Exercises . . . . . 120
<b>4</b>	<b>Reasoning with Constraints 125</b>
4.1	Possible Worlds, Variables, and Constraints . . . . . 125
4.1.1	Variables and Worlds . . . . . 125
4.1.2	Constraints . . . . . 129
4.1.3	Constraint Satisfaction Problems . . . . . 131
4.2	Generate-and-Test Algorithms . . . . . 132
4.3	Solving CSPs Using Search . . . . . 133
4.4	Consistency Algorithms . . . . . 134
4.5	Domain Splitting . . . . . 139
4.6	Variable Elimination . . . . . 141
4.7	Local Search . . . . . 144
4.7.1	Iterative Best Improvement . . . . . 146
4.7.2	Randomized Algorithms . . . . . 148
4.7.3	Local Search Variants . . . . . 149
4.7.4	Evaluating Randomized Algorithms . . . . . 153
4.7.5	Random Restart . . . . . 156
4.8	Population-Based Methods . . . . . 156
4.9	Optimization . . . . . 160
4.9.1	Systematic Methods for Optimization . . . . . 162
4.9.2	Local Search for Optimization . . . . . 164
4.10	Review . . . . . 167
4.11	References and Further Reading . . . . . 167
4.12	Exercises . . . . . 168
<b>5</b>	<b>Propositions and Inference 173</b>
5.1	Propositions . . . . . 173
5.1.1	Syntax of Propositional Calculus . . . . . 174

5.1.2	Semantics of the Propositional Calculus . . . . .	175
5.2	Propositional Constraints . . . . .	179
5.2.1	Clausal Form for Consistency Algorithms . . . . .	180
5.2.2	Exploiting Propositional Structure in Local Search . . . . .	181
5.3	Propositional Definite Clauses . . . . .	182
5.3.1	Questions and Answers . . . . .	185
5.3.2	Proofs . . . . .	186
5.4	Knowledge Representation Issues . . . . .	194
5.4.1	Background Knowledge and Observations . . . . .	194
5.4.2	Querying the User . . . . .	194
5.4.3	Knowledge-Level Explanation . . . . .	196
5.4.4	Knowledge-Level Debugging . . . . .	199
5.5	Proving by Contradiction . . . . .	204
5.5.1	Horn Clauses . . . . .	205
5.5.2	Assumables and Conflicts . . . . .	206
5.5.3	Consistency-Based Diagnosis . . . . .	207
5.5.4	Reasoning with Assumptions and Horn Clauses . . . . .	209
5.6	Complete Knowledge Assumption . . . . .	212
5.6.1	Non-monotonic Reasoning . . . . .	216
5.6.2	Proof Procedures for Negation as Failure . . . . .	217
5.7	Abduction . . . . .	220
5.8	Causal Models . . . . .	225
5.9	Review . . . . .	226
5.10	References and Further Reading . . . . .	227
5.11	Exercises . . . . .	228
<b>6</b>	<b>Planning with Certainty</b> . . . . .	<b>239</b>
6.1	Representing States, Actions, and Goals . . . . .	240
6.1.1	Explicit State-Space Representation . . . . .	241
6.1.2	The STRIPS Representation . . . . .	243
6.1.3	Feature-Based Representation of Actions . . . . .	244
6.1.4	Initial States and Goals . . . . .	246
6.2	Forward Planning . . . . .	246
6.3	Regression Planning . . . . .	249
6.4	Planning as a CSP . . . . .	252
6.4.1	Action Features . . . . .	255
6.5	Partial-Order Planning . . . . .	257
6.6	Review . . . . .	260
6.7	References and Further Reading . . . . .	261
6.8	Exercises . . . . .	262
<b>7</b>	<b>Supervised Machine Learning</b> . . . . .	<b>267</b>
7.1	Learning Issues . . . . .	268

Contents	ix
7.2	Supervised Learning . . . . . 271
7.2.1	Evaluating Predictions . . . . . 274
7.2.2	Types of Errors . . . . . 279
7.2.3	Point Estimates with No Input Features . . . . . 283
7.3	Basic Models for Supervised Learning . . . . . 285
7.3.1	Learning Decision Trees . . . . . 285
7.3.2	Linear Regression and Classification . . . . . 291
7.4	Overfitting . . . . . 298
7.4.1	Pseudocounts . . . . . 301
7.4.2	Regularization . . . . . 304
7.4.3	Cross Validation . . . . . 306
7.5	Neural Networks and Deep Learning . . . . . 308
7.6	Composite Models . . . . . 316
7.6.1	Random Forests . . . . . 317
7.6.2	Ensemble Learning . . . . . 318
7.7	Case-Based Reasoning . . . . . 320
7.8	Learning as Refining the Hypothesis Space . . . . . 323
7.8.1	Version-Space Learning . . . . . 325
7.8.2	Probably Approximately Correct Learning . . . . . 328
7.9	Review . . . . . 331
7.10	References and Further Reading . . . . . 331
7.11	Exercises . . . . . 333
<b>III Reasoning, Learning and Acting with Uncertainty</b>	<b>341</b>
<b>8 Reasoning with Uncertainty</b>	<b>343</b>
8.1	Probability . . . . . 343
8.1.1	Semantics of Probability . . . . . 345
8.1.2	Axioms for Probability . . . . . 347
8.1.3	Conditional Probability . . . . . 350
8.1.4	Expected Values . . . . . 355
8.1.5	Information . . . . . 356
8.2	Independence . . . . . 358
8.3	Belief Networks . . . . . 360
8.3.1	Observations and Queries . . . . . 362
8.3.2	Constructing Belief Networks . . . . . 363
8.4	Probabilistic Inference . . . . . 370
8.4.1	Variable Elimination for Belief Networks . . . . . 372
8.4.2	Representing Conditional Probabilities and Factors . . . 381
8.5	Sequential Probability Models . . . . . 384
8.5.1	Markov Chains . . . . . 384
8.5.2	Hidden Markov Models . . . . . 387



8.5.3	Algorithms for Monitoring and Smoothing . . . . .	392
8.5.4	Dynamic Belief Networks . . . . .	393
8.5.5	Time Granularity . . . . .	394
8.5.6	Probabilistic Models of Language . . . . .	395
8.6	Stochastic Simulation . . . . .	402
8.6.1	Sampling from a Single Variable . . . . .	403
8.6.2	Forward Sampling in Belief Networks . . . . .	404
8.6.3	Rejection Sampling . . . . .	405
8.6.4	Likelihood Weighting . . . . .	407
8.6.5	Importance Sampling . . . . .	408
8.6.6	Particle Filtering . . . . .	410
8.6.7	Markov Chain Monte Carlo . . . . .	412
8.7	Review . . . . .	414
8.8	References and Further Reading . . . . .	414
8.9	Exercises . . . . .	415
<b>9</b>	<b>Planning with Uncertainty</b>	<b>425</b>
9.1	Preferences and Utility . . . . .	426
9.1.1	Axioms for Rationality . . . . .	426
9.1.2	Factored Utility . . . . .	433
9.1.3	Prospect Theory . . . . .	435
9.2	One-Off Decisions . . . . .	438
9.2.1	Single-Stage Decision Networks . . . . .	442
9.3	Sequential Decisions . . . . .	444
9.3.1	Decision Networks . . . . .	445
9.3.2	Policies . . . . .	449
9.3.3	Variable Elimination for Decision Networks . . . . .	451
9.4	The Value of Information and Control . . . . .	455
9.5	Decision Processes . . . . .	458
9.5.1	Policies . . . . .	462
9.5.2	Value Iteration . . . . .	464
9.5.3	Policy Iteration . . . . .	468
9.5.4	Dynamic Decision Networks . . . . .	470
9.5.5	Partially Observable Decision Processes . . . . .	474
9.6	Review . . . . .	475
9.7	References and Further Reading . . . . .	476
9.8	Exercises . . . . .	476
<b>10</b>	<b>Learning with Uncertainty</b>	<b>487</b>
10.1	Probabilistic Learning . . . . .	487
10.1.1	Learning Probabilities . . . . .	488
10.1.2	Probabilistic Classifiers . . . . .	491
10.1.3	MAP Learning of Decision Trees . . . . .	496

Contents	xi
10.1.4 Description Length . . . . .	498
10.2 Unsupervised Learning . . . . .	499
10.2.1 $k$ -Means . . . . .	499
10.2.2 Expectation Maximization for Soft Clustering . . . . .	503
10.3 Learning Belief Networks . . . . .	507
10.3.1 Learning the Probabilities . . . . .	508
10.3.2 Hidden Variables . . . . .	509
10.3.3 Missing Data . . . . .	509
10.3.4 Structure Learning . . . . .	510
10.3.5 General Case of Belief Network Learning . . . . .	512
10.4 Bayesian Learning . . . . .	512
10.5 Review . . . . .	517
10.6 References and Further Reading . . . . .	518
10.7 Exercises . . . . .	518
<b>11 Multiagent Systems</b>	<b>521</b>
11.1 Multiagent Framework . . . . .	521
11.2 Representations of Games . . . . .	523
11.2.1 Normal Form Games . . . . .	523
11.2.2 Extensive Form of a Game . . . . .	524
11.2.3 Multiagent Decision Networks . . . . .	527
11.3 Computing Strategies with Perfect Information . . . . .	528
11.4 Reasoning with Imperfect Information . . . . .	532
11.4.1 Computing Nash Equilibria . . . . .	538
11.5 Group Decision Making . . . . .	541
11.6 Mechanism Design . . . . .	542
11.7 Review . . . . .	544
11.8 References and Further Reading . . . . .	545
11.9 Exercises . . . . .	545
<b>12 Learning to Act</b>	<b>549</b>
12.1 Reinforcement Learning Problem . . . . .	549
12.2 Evolutionary Algorithms . . . . .	553
12.3 Temporal Differences . . . . .	554
12.4 Q-learning . . . . .	555
12.5 Exploration and Exploitation . . . . .	557
12.6 Evaluating Reinforcement Learning Algorithms . . . . .	559
12.7 On-Policy Learning . . . . .	560
12.8 Model-Based Reinforcement Learning . . . . .	562
12.9 Reinforcement Learning with Features . . . . .	565
12.9.1 SARSA with Linear Function Approximation . . . . .	565
12.10 Multiagent Reinforcement Learning . . . . .	569
12.10.1 Perfect-Information Games . . . . .	569

12.10.2 Learning to Coordinate . . . . .	569
12.11 Review . . . . .	574
12.12 References and Further Reading . . . . .	574
12.13 Exercises . . . . .	575
<b>IV Reasoning, Learning and Acting with Individuals and Relations</b>	<b>579</b>
<b>13 Individuals and Relations</b>	<b>581</b>
13.1 Exploiting Relational Structure . . . . .	582
13.2 Symbols and Semantics . . . . .	583
13.3 Datalog: A Relational Rule Language . . . . .	584
13.3.1 Semantics of Ground Datalog . . . . .	587
13.3.2 Interpreting Variables . . . . .	589
13.3.3 Queries with Variables . . . . .	595
13.4 Proofs and Substitutions . . . . .	597
13.4.1 Instances and Substitutions . . . . .	597
13.4.2 Bottom-up Procedure with Variables . . . . .	599
13.4.3 Unification . . . . .	601
13.4.4 Definite Resolution with Variables . . . . .	602
13.5 Function Symbols . . . . .	604
13.5.1 Proof Procedures with Function Symbols . . . . .	610
13.6 Applications in Natural Language . . . . .	612
13.6.1 Using Definite Clauses for Context-Free Grammars . . . . .	614
13.6.2 Augmenting the Grammar . . . . .	618
13.6.3 Building Structures for Non-terminals . . . . .	619
13.6.4 Canned Text Output . . . . .	619
13.6.5 Enforcing Constraints . . . . .	620
13.6.6 Building a Natural Language Interface to a Database . . . . .	621
13.6.7 Limitations . . . . .	627
13.7 Equality . . . . .	628
13.7.1 Allowing Equality Assertions . . . . .	629
13.7.2 Unique Names Assumption . . . . .	630
13.8 Complete Knowledge Assumption . . . . .	633
13.8.1 Complete Knowledge Assumption Proof Procedures . . . . .	637
13.9 Review . . . . .	638
13.10 References and Further Reading . . . . .	638
13.11 Exercises . . . . .	639
<b>14 Ontologies and Knowledge-Based Systems</b>	<b>645</b>
14.1 Knowledge Sharing . . . . .	645
14.2 Flexible Representations . . . . .	646

Contents	xiii
14.2.1 Choosing Individuals and Relations . . . . .	647
14.2.2 Graphical Representations . . . . .	650
14.2.3 Classes . . . . .	652
14.3 Ontologies and Knowledge Sharing . . . . .	655
14.3.1 Uniform Resource Identifiers . . . . .	661
14.3.2 Description Logic . . . . .	662
14.3.3 Top-Level Ontologies . . . . .	670
14.4 Implementing Knowledge-Based Systems . . . . .	673
14.4.1 Base Languages and Metalanguages . . . . .	674
14.4.2 A Vanilla Meta-Interpreter . . . . .	676
14.4.3 Expanding the Base Language . . . . .	678
14.4.4 Depth-Bounded Search . . . . .	680
14.4.5 Meta-Interpreter to Build Proof Trees . . . . .	681
14.4.6 Delaying Goals . . . . .	682
14.5 Review . . . . .	684
14.6 References and Further Reading . . . . .	684
14.7 Exercises . . . . .	685
<b>15 Relational Planning, Learning, and Probabilistic Reasoning</b>	<b>691</b>
15.1 Planning with Individuals and Relations . . . . .	692
15.1.1 Situation Calculus . . . . .	692
15.1.2 Event Calculus . . . . .	699
15.2 Relational Learning . . . . .	701
15.2.1 Structure Learning: Inductive Logic Programming . . . . .	701
15.2.2 Learning Hidden Properties: Collaborative Filtering . . . . .	706
15.3 Statistical Relational Artificial Intelligence . . . . .	711
15.3.1 Relational Probabilistic Models . . . . .	711
15.4 Review . . . . .	724
15.5 References and Further Reading . . . . .	724
15.6 Exercises . . . . .	725
<b>V Retrospect and Prospect</b>	<b>729</b>
<b>16 Retrospect and Prospect</b>	<b>731</b>
16.1 Dimensions of Complexity Revisited . . . . .	731
16.2 Social and Ethical Consequences . . . . .	736
16.3 References and Further Reading . . . . .	742
16.4 Exercises . . . . .	742
<b>A Mathematical Preliminaries and Notation</b>	<b>745</b>
A.1 Discrete Mathematics . . . . .	745
A.2 Functions, Factors and Arrays . . . . .	746

xiv	Contents
A.3	Relations and the Relational Algebra . . . . . 747
<b>References</b>	<b>751</b>
<b>Index</b>	<b>773</b>

# Figures

1	Overview of chapters and dependencies . . . . .	xxvii
1.1	Part of Turing's possible dialog for the Turing test . . . . .	6
1.2	Some questions CHAT-80 could answer . . . . .	10
1.3	An agent interacting with an environment . . . . .	12
1.4	The role of representations in solving tasks . . . . .	16
1.5	Solution quality as a function of time for an anytime algorithm	26
1.6	Dimensions of complexity . . . . .	31
1.7	A typical laboratory environment for the delivery robot . . . .	36
1.8	An electrical environment for the diagnostic assistant . . . . .	38
2.1	An agent system and its components . . . . .	50
2.2	Percept traces for Example 2.1 . . . . .	53
2.3	Command trace for Example 2.1 (page 52) . . . . .	53
2.4	An idealized hierarchical agent system architecture . . . . .	57
2.5	A hierarchical decomposition of the delivery robot . . . . .	60
2.6	The middle layer of the delivery robot . . . . .	62
2.7	The top layer of the delivery robot controller . . . . .	63
2.8	A simulation of the robot carrying out the plan of Example 2.6	64
2.9	Offline and online decomposition of an agent . . . . .	66
2.10	Internals of an agent, showing roles . . . . .	68
2.11	A robot trap . . . . .	73
3.1	The delivery robot domain with interesting locations labeled .	80
3.2	A graph with arc costs for the delivery robot domain . . . . .	83
3.3	Problem solving by graph searching . . . . .	85

3.4	Search: generic graph searching algorithm . . . . .	86
3.5	The order in which paths are expanded in breadth-first search . . . . .	89
3.6	The order paths are expanded in depth-first search . . . . .	90
3.7	A graph, with cycles, for the delivery robot domain . . . . .	94
3.8	<i>ID_search</i> : iterative deepening search . . . . .	96
3.9	A graph that is bad for greedy best-first search . . . . .	100
3.10	Triangle inequality: $h(n) \leq cost(n, n') + h(n')$ . . . . .	107
3.11	Summary of search strategies . . . . .	109
3.12	Depth-first branch-and-bound search . . . . .	111
3.13	The paths expanded in depth-first branch-and-bound search . . . . .	112
3.14	A grid-searching problem . . . . .	120
4.1	Search tree for the CSP of Example 4.12 . . . . .	134
4.2	Constraint network for the CSP of Example 4.14 . . . . .	135
4.3	Generalized arc consistency algorithm . . . . .	136
4.4	Domain-consistent constraint network . . . . .	138
4.5	Finding a model for a CSP using arc consistency and domain splitting . . . . .	141
4.6	Variable elimination for finding all solutions to a CSP . . . . .	143
4.7	Local search for finding a solution to a CSP . . . . .	145
4.8	Two search spaces; find the minimum . . . . .	148
4.9	Probability of simulated annealing accepting worsening steps . . . . .	153
4.10	Run-time distributions . . . . .	155
4.11	Genetic algorithm for finding a solution to a CSP . . . . .	158
4.12	Variable elimination for optimizing with soft constraints . . . . .	163
4.13	Gradient descent . . . . .	166
4.14	A crossword puzzle to be solved with six words . . . . .	168
4.15	Two crossword puzzles . . . . .	169
4.16	Abstract constraint network . . . . .	172
5.1	Truth table defining $\neg$ , $\wedge$ , $\vee$ , $\leftarrow$ , $\rightarrow$ , and $\leftrightarrow$ . . . . .	175
5.2	An electrical environment with components named . . . . .	183
5.3	Bottom-up proof procedure for computing consequences of <i>KB</i> . . . . .	187
5.4	Top-down definite clause proof procedure . . . . .	191
5.5	A search graph for a top-down derivation . . . . .	192
5.6	An algorithm to debug false positive answers . . . . .	201
5.7	An algorithm for debugging missing answers (false negatives) . . . . .	203
5.8	Knowledge for Example 5.23 (page 207) . . . . .	208
5.9	Bottom-up proof procedure for computing conflicts . . . . .	210
5.10	Top-down Horn clause interpreter to find conflicts . . . . .	212
5.11	Bottom-up negation as failure proof procedure . . . . .	218
5.12	Top-down negation as failure interpreter . . . . .	219
5.13	The plumbing domain . . . . .	229

Figures	xvii
5.14	Deep Space One engine design . . . . . 233
5.15	A space communication network . . . . . 237
6.1	The delivery robot domain . . . . . 240
6.2	Part of the search space for a state-space planner . . . . . 247
6.3	Part of the search space for a regression planner . . . . . 250
6.4	The delivery robot CSP planner for a planning horizon of $k = 2$ . . . . . 254
6.5	The delivery robot CSP planner with factored actions . . . . . 256
6.6	Partial-order planner . . . . . 259
7.1	Examples of a user's preferences . . . . . 273
7.2	Training and test examples for a regression task . . . . . 274
7.3	Linear regression predictions for a simple prediction example. . . . . 277
7.4	Six predictors in the ROC space and the precision-recall space . . . . . 282
7.5	Optimal prediction for binary classification . . . . . 284
7.6	Two decision trees . . . . . 286
7.7	Decision tree learner . . . . . 287
7.8	Incremental gradient descent for learning a linear function . . . . . 293
7.9	The sigmoid or logistic function . . . . . 294
7.10	Stochastic gradient descent for logistic regression . . . . . 296
7.11	Linear separators for Boolean functions . . . . . 297
7.12	Predicting what holiday a person likes . . . . . 298
7.13	Fitting polynomials to the data of Figure 7.2 (page 274) . . . . . 300
7.14	Error as a function of number of steps . . . . . 302
7.15	Validation error and test set error . . . . . 309
7.16	A deep neural network . . . . . 311
7.17	Back-propagation for a multilayer neural network . . . . . 314
7.18	A neural network for Example 7.20 (page 313) . . . . . 315
7.19	Functional gradient boosting regression learner . . . . . 319
7.20	Error of functional gradient boosting of decision trees . . . . . 320
7.21	Candidate elimination algorithm . . . . . 327
7.22	Training examples for Exercise 7.3 . . . . . 334
8.1	Ten worlds described by variables <i>Filled</i> and <i>Shape</i> . . . . . 346
8.2	Belief network for exam answering of Example 8.13 . . . . . 362
8.3	Belief network for report of leaving of Example 8.15 . . . . . 365
8.4	Belief network for Example 8.16 . . . . . 366
8.5	Belief network for the electrical domain of Figure 1.8 . . . . . 368
8.6	Conditional probability tables . . . . . 373
8.7	An example factor and assignments . . . . . 375
8.8	Multiplying factors . . . . . 375
8.9	Summing out a variable from a factor . . . . . 376
8.10	Variable elimination for belief networks . . . . . 378



8.11	A Markov chain as a belief network . . . . .	384
8.12	A hidden Markov model as a belief network . . . . .	387
8.13	A belief network for localization . . . . .	389
8.14	Localization domain . . . . .	390
8.15	A distribution over locations . . . . .	391
8.16	Localization with multiple sensors . . . . .	391
8.17	Two-stage dynamic belief network for paper pricing . . . . .	394
8.18	Expanded dynamic belief network for paper pricing . . . . .	395
8.19	Set-of-words language model . . . . .	396
8.20	Naive belief network with a set-of-words model for a help system . . . . .	397
8.21	Bag-of-words or unigram language model . . . . .	398
8.22	Bigram language model . . . . .	398
8.23	Trigram language model . . . . .	399
8.24	Some of the most-likely $n$ -grams . . . . .	399
8.25	Predictive typing model . . . . .	400
8.26	Simple topic model with a set-of-words . . . . .	401
8.27	A cumulative probability distribution . . . . .	403
8.28	Sampling for a belief network . . . . .	405
8.29	Rejection sampling for $P(\textit{tampering} \mid \textit{smoke} \wedge \neg \textit{report})$ . . . . .	406
8.30	Likelihood weighting for belief network inference . . . . .	408
8.31	Particle filtering for belief network inference . . . . .	411
8.32	Gibbs sampling for belief network inference . . . . .	413
8.33	Counts for students in departments . . . . .	416
8.34	A simple diagnostic belief network . . . . .	416
8.35	Belief network for an overhead projector . . . . .	417
8.36	Belief network for a nuclear submarine . . . . .	421
9.1	The preference between $o_2$ and the lottery, as a function of $p$ . . . . .	429
9.2	Money–utility relationships for agents with different risk profiles . . . . .	432
9.3	Possible money–utility relationship from Example 9.2 . . . . .	432
9.4	Human perception of length depends on the context . . . . .	436
9.5	Money–value relationship for prospect theory . . . . .	436
9.6	A decision tree for the delivery robot . . . . .	440
9.7	Single-stage decision network for the delivery robot . . . . .	443
9.8	Variable elimination for a single-stage decision network . . . . .	444
9.9	Decision network for decision of whether to take an umbrella . . . . .	446
9.10	Decision network for idealized test-treat diagnosis scenario . . . . .	447
9.11	Decision network for the fire alarm decision problem . . . . .	448
9.12	Utility for fire alarm decision network . . . . .	448
9.13	Variable elimination for decision networks . . . . .	451
9.14	Decision network representing a finite part of an MDP . . . . .	459
9.15	The grid world of Example 9.28 (page 460) . . . . .	460

Figures	xix
9.16	Value iteration for MDPs, storing $V$ . . . . . 465
9.17	Asynchronous value iteration for MDPs . . . . . 468
9.18	Policy iteration for MDPs . . . . . 469
9.19	Two-stage dynamic decision network . . . . . 472
9.20	Dynamic decision network unfolded for a horizon of 3 . . . . . 473
9.21	Dynamic decision network with intermediate variables for a horizon of 2, omitting the reward nodes . . . . . 473
9.22	A POMDP as a dynamic decision network . . . . . 475
9.23	A decision network for an invitation decision . . . . . 477
9.24	Utility function for the study decision . . . . . 478
9.25	Decision about whether to cheat . . . . . 479
10.1	Belief network corresponding to a naive Bayes classifier . . . . . 491
10.2	$k$ -means for unsupervised learning . . . . . 501
10.3	A trace of the $k$ -means algorithm for $k = 2$ for Example 10.9 . . . . . 502
10.4	EM algorithm: Bayes classifier with hidden class . . . . . 504
10.5	EM algorithm for unsupervised learning . . . . . 505
10.6	EM for unsupervised learning . . . . . 506
10.7	From the model and the data, learn the probabilities . . . . . 508
10.8	Deriving probabilities with missing data . . . . . 509
10.9	EM algorithm for belief networks with hidden variables . . . . . 510
10.10	The i.i.d. assumption as a belief network . . . . . 513
10.11	Beta distribution based on different samples . . . . . 515
11.1	Normal form for the rock-paper-scissors game . . . . . 524
11.2	Extensive form of the sharing game . . . . . 525
11.3	Extensive form of the rock-paper-scissors game . . . . . 527
11.4	Multiagent decision network for Example 11.5 . . . . . 528
11.5	A zero-sum game tree showing which nodes can be pruned . . . . . 530
11.6	Minimax with $\alpha$ - $\beta$ pruning . . . . . 531
11.7	Soccer penalty kick . . . . . 533
11.8	Probability of a goal as a function of action probabilities . . . . . 534
12.1	The environment of a tiny reinforcement learning problem . . . . . 550
12.2	The environment of a grid game . . . . . 551
12.3	Q-learning controller . . . . . 557
12.4	Cumulative reward as a function of the number of steps . . . . . 559
12.5	SARSA: on-policy reinforcement learning . . . . . 561
12.6	Model-based reinforcement learner . . . . . 564
12.7	SARSA with linear function approximation . . . . . 567
12.8	Learning to coordinate . . . . . 570
12.9	Learning for the football–shopping coordination example . . . . . 572
12.10	Learning for the soccer penalty kick example . . . . . 573

13.1	The role of semantics . . . . .	584
13.2	A knowledge base about rooms . . . . .	596
13.3	Unification algorithm for Datalog . . . . .	601
13.4	Top-down definite clause proof procedure for Datalog . . . . .	604
13.5	A derivation for query <i>two_doors_east(R, r107)</i> . . . . .	605
13.6	Axiomatizing a “before” relation for dates in the common era . . . . .	607
13.7	A context-free grammar for a restricted subset of English . . . . .	617
13.8	A simple dictionary . . . . .	618
13.9	Grammar for output of canned English . . . . .	620
13.10	Grammar to enforce number agreement and build a parse tree . . . . .	622
13.11	A grammar that directly answers a question . . . . .	624
13.12	A grammar that constructs a query . . . . .	626
13.13	A dictionary for constructing a query . . . . .	627
13.14	Two chairs . . . . .	629
14.1	A semantic network . . . . .	651
14.2	A semantic network allowing inheritance . . . . .	654
14.3	Mapping from a conceptualization to a symbol . . . . .	659
14.4	Some OWL built-in classes and class constructors . . . . .	663
14.5	Some RDF, RDF-S, and OWL built-in predicates . . . . .	664
14.6	OWL functional syntax representation of Example 14.14 . . . . .	666
14.7	Categorizing an entity in a top-level ontology . . . . .	671
14.8	The non-ground representation for the base language . . . . .	675
14.9	The vanilla definite clause meta-interpreter . . . . .	676
14.10	A knowledge base for house wiring . . . . .	677
14.11	A meta-interpreter that uses built-in calls and disjunction . . . . .	679
14.12	A meta-interpreter for depth-bounded search . . . . .	680
14.13	A meta-interpreter that builds a proof tree . . . . .	681
14.14	A meta-interpreter that collects delayed goals . . . . .	683
15.1	Data about holiday preferences . . . . .	702
15.2	Top-down induction of a logic program . . . . .	705
15.3	Part of the MovieLens data set . . . . .	707
15.4	Movie ratings by users as a function of a single property . . . . .	710
15.5	Gradient descent for collaborative filtering . . . . .	712
15.6	Predict which student will do better in course $c_4$ . . . . .	713
15.7	Belief network for two-digit addition . . . . .	714
15.8	A plate model to predict the grades of students . . . . .	716
15.9	A grounding for 3 students and 2 courses . . . . .	717
15.10	A grounding that is sufficient to predict from the data in Figure 15.6 . . . . .	718
15.11	Belief network with plates for multidigit addition . . . . .	718

Figures	xxi
16.1 Some representations rated by dimensions of complexity . . .	732

# Preface

*Artificial Intelligence: Foundations of Computational Agents* is a book about the science of artificial intelligence (AI). AI is the study of the design of intelligent computational agents. The book is structured as a textbook but it is designed to be accessible to a wide audience.

We wrote this book because we are excited about the emergence of AI as an integrated science. As with any science being developed, AI has a coherent, formal theory and a rambunctious experimental wing. Here we balance theory and experiment and show how to link them together intimately. We develop the science of AI together with its engineering applications. We believe the adage, “There is nothing so practical as a good theory.” The spirit of our approach is captured by the dictum, “Everything should be made as simple as possible, but not simpler.” We must build the science on solid foundations; we present the foundations, but only sketch, and give some examples of, the complexity required to build useful intelligent systems. Although the resulting systems will be complex, the foundations and the building blocks should be simple.

This second edition results from extensive revision throughout the text. We have restructured the material based on feedback from instructors who have used the book in classes. We have brought it up to date to reflect the current state of the art, made parts that were difficult for students more straightforward, added more intuitive explanations, and coordinated the pseudocode algorithms with new open-source implementations of the algorithms in Python and Prolog. We have resisted the temptation to just keep adding more material. AI research is expanding so rapidly now that the volume of potential new text material is vast. However, research teaches us not only what works but

also what does not work so well, allowing us to be highly selective. We have included more material on machine learning techniques that have proven successful. However, research also has trends and fashions. We have removed techniques that have been shown to be less promising, but we distinguish them from the techniques that are merely out of fashion. We include some currently unfashionable material if the problems attacked still remain and the techniques have the potential to form the basis for future research and development. We have further developed the concept of a single design space for intelligent agents, showing how many bewilderingly diverse techniques can be seen in a simple, uniform framework. This allows us to emphasize the principles underlying the foundations of computational agents, making those ideas more accessible to students.

The book can be used as an introductory text on artificial intelligence for advanced undergraduate or graduate students in computer science or related disciplines such as computer engineering, philosophy, cognitive science, or psychology. It will appeal more to the technically minded; parts are technically challenging, focusing on learning by doing: designing, building, and implementing systems. Any curious scientifically oriented reader will benefit from studying the book. Previous experience with computational systems is desirable, but prior study of the foundations upon which we build, including logic, probability, calculus, and control theory, is not necessary, because we develop the concepts as required.

The serious student will gain valuable skills at several levels ranging from expertise in the specification and design of intelligent agents to skills for implementing, testing, and improving real software systems for several challenging application domains. The thrill of participating in the emergence of a new science of intelligent agents is one of the attractions of this approach. The practical skills of dealing with a world of ubiquitous, intelligent, embedded agents are now in great demand in the marketplace.

The focus is on an intelligent agent acting in an environment. We start with simple agents acting in simple, static environments and gradually increase the power of the agents to cope with more challenging worlds. We explore ten dimensions of complexity that allow us to introduce, gradually and with modularity, what makes building intelligent agents challenging. We have tried to structure the book so that the reader can understand each of the dimensions separately and we make this concrete by repeatedly illustrating the ideas with four different agent tasks: a delivery robot, a diagnostic assistant, a tutoring system, and a trading agent.

The agent we want the student to envision is a hierarchically designed agent that acts intelligently in a stochastic environment that it can only partially observe – one that reasons online about individuals and relationships among them, has complex preferences, learns while acting, takes into account

other agents, and acts appropriately given its own computational limitations. Of course, we cannot start with such an agent; it is still a research question to build such agents. So we introduce the simplest agents and then show how to add each of these complexities in a modular way.

We have made a number of design choices which distinguish this book from competing books, including our earlier book.

- We have tried to give a coherent framework in which to understand AI. We have chosen not to present disconnected topics that do not fit together. For example, we do not present disconnected logical and probabilistic views of AI, but we have presented a multidimensional design space in which the students can understand the big picture, in which probabilistic and logical reasoning coexist.
- We decided that it is better to clearly explain the foundations upon which more sophisticated techniques can be built, rather than present these more sophisticated techniques. This means that a larger gap may exist between what is covered in this book and the frontier of science. But it also means that the student will have a better foundation to understand current and future research.
- One of the more difficult decisions we made was how to linearize the design space. Our previous book [Poole et al., 1998] presented a relational language early and built the foundations in terms of this language. This approach made it difficult for the students to appreciate work that was not relational, for example, in reinforcement learning that is developed in terms of states. In this book, we have chosen a relations-late approach. This approach probably reflects better the research over the past few decades where there has been much progress in reasoning and learning for feature-based representations. It also allows the student to understand that probabilistic and logical reasoning are complementary. The book, however, is structured so that an instructor could present relations earlier.

We provide open-source Python implementations of the algorithms (<http://www.aipython.org>); these are designed to be useful and to highlight the main ideas without extra frills to interfere with the main ideas. This book uses examples from Alspace.org (<http://www.aispace.org>), a collection of pedagogical applets that we have been involved in designing. To gain further experience in building intelligent systems, a student should also experiment with a high-level symbol-manipulation language, such as Haskell, Lisp or Prolog. We also provide implementations in AILog, a clean logic programming language related to Prolog, designed to demonstrate many of the issues in this book. These

tools are intended to be helpful, but not essential to an understanding or use of the ideas in this book.

Our approach, through the development of the power of the agent's capabilities and representation language, is both simpler and more powerful than the traditional approach of surveying and cataloging various applications of AI. However, as a consequence, some applications such as the details of computational vision or computational linguistics are not covered in this book.

We have chosen not to present an encyclopedic view of AI. Not every major idea that has been investigated is presented here. We have chosen some basic ideas upon which other, more sophisticated, techniques are based and have tried to explain the basic ideas in detail, sketching how these can be expanded.

Figure 1 (page xxvii) shows the topics covered in the book. The solid lines depict prerequisites. Often the prerequisite structure does not include all sub-topics. Given the medium of a book, we have had to linearize the topics. However, the book is designed so the topics are teachable in any order satisfying the prerequisite structure.

The references given at the end of each chapter are not meant to be comprehensive; we have referenced works that we have directly used and works that we think provide good overviews of the literature, by referencing both classic works and more recent surveys. We hope that no researchers feel slighted by their omission, and we are happy to have feedback where someone feels that an idea has been misattributed. Remember that this book is *not* a survey of AI research.

We invite you to join us in an intellectual adventure: building a science of intelligent agents.

David Poole  
Alan Mackworth

## Acknowledgments

Thanks to Randy Goebel for valuable input on this book. We also gratefully acknowledge the helpful comments on the first edition and earlier drafts of the second edition received from Guy van den Broeck, David Buchman, Giuseppe Carenini, Cristina Conati, Mark Crowley, Matthew Dirks, Bahare Fatemi, Pooyan Fazli, Robert Holte, Holger Hoos, Manfred Jaeger, Mehran Kazemi, Mohammad Reza Khojasteh, Jacek Kiszyński, Richard Korf, Bob Kowalski, Kevin Leyton-Brown, Josje Lodder, Marian Mackworth, Gabriel Murray, Sriraam Nataraajan, Alex Poole, Alessandro Proveti, Mark Schmidt, Marco Valtorta, and the anonymous reviewers. Thanks to the students who pointed out many errors in the earlier drafts. Thanks to Jen Fernquist for the website design. David would like to thank Thomas Lukasiewicz and The Leverhulme Trust for sponsoring



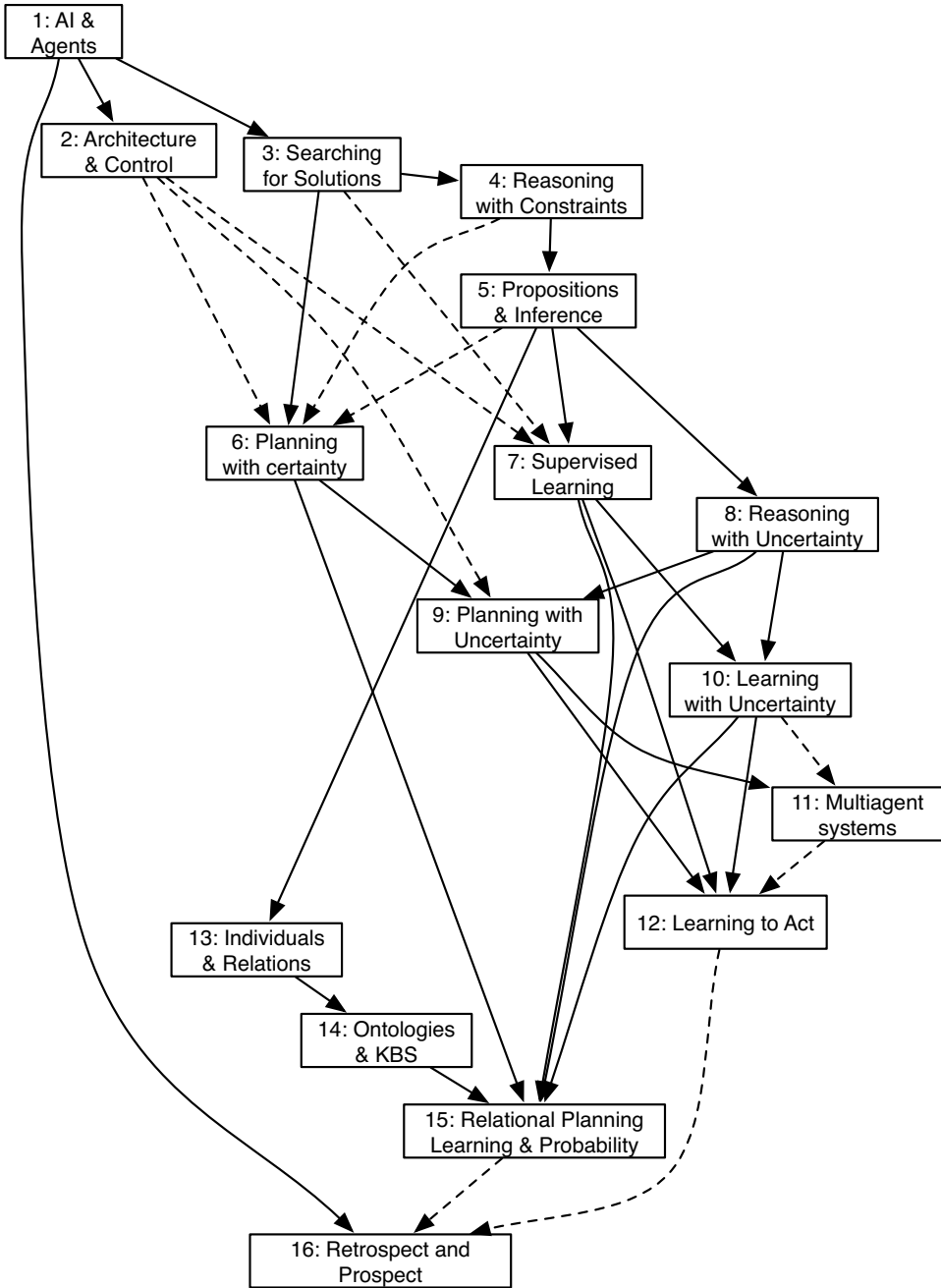


Figure 1: Overview of chapters and dependencies

his sabbatical in Oxford, where much of this second edition was written. We are grateful to James Falen for permission to quote his poem on constraints.

The quote at the beginning of Chapter 9 is reprinted with permission of Simon & Schuster, Inc. from *THE CREATIVE HABIT: Learn it and Use It* by Twyla Tharp with Mark Reiter. Copyright 2003 by W.A.T. Ltd. All Rights Reserved.

Thanks to our editor Lauren Cowles and the staff at Cambridge University Press for all their support, encouragement and help. All the mistakes remaining are ours.