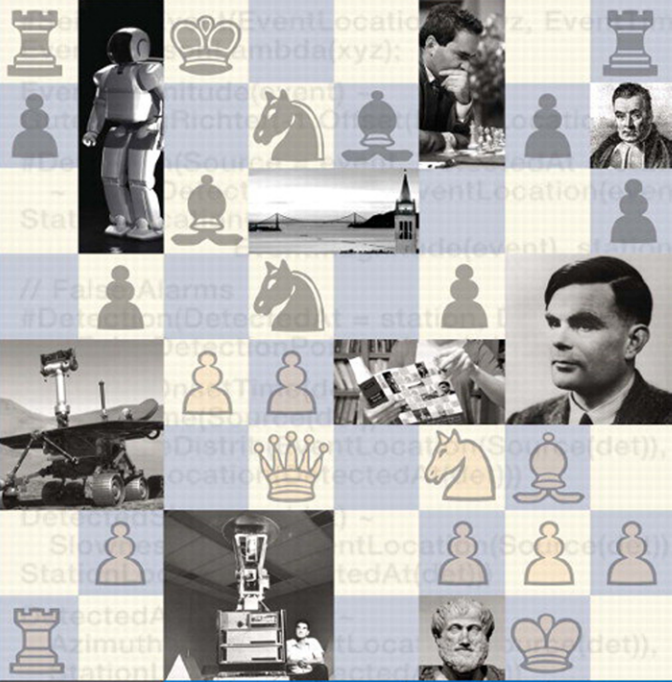


Stuart
Russell
Peter
Norvig

Artificial Intelligence
A Modern Approach
Third Edition

Artificial Intelligence

A Modern Approach

Third Edition

PRENTICE HALL SERIES
IN ARTIFICIAL INTELLIGENCE
Stuart Russell and Peter Norvig, Editors

FORSYTH & PONCE	*Computer Vision: A Modern Approach*
GRAHAM	ANSI *Common Lisp*
JURAFSKY & MARTIN	*Speech and Language Processing, 2nd ed.*
NEAPOLITAN	*Learning Bayesian Networks*
RUSSELL & NORVIG	*Artificial Intelligence: A Modern Approach, 3rd ed.*

Artificial Intelligence

A Modern Approach

Third Edition

Stuart J. Russell and Peter Norvig

Contributing writers:
Ernest Davis
Douglas D. Edwards
David Forsyth
Nicholas J. Hay
Jitendra M. Malik
Vibhu Mittal
Mehran Sahami
Sebastian Thrun

Prentice Hall
Upper Saddle River Boston Columbus San Francisco New York
Indianapolis London Toronto Sydney Singapore Tokyo Montreal
Dubai Madrid Hong Kong Mexico City Munich Paris Amsterdam Cape Town

Vice President and Editorial Director, ECS: Marcia J. Horton
Editor-in-Chief: Michael Hirsch
Executive Editor: Tracy Dunkelberger
Assistant Editor: Melinda Haggerty
Editorial Assistant: Allison Michael
Vice President, Production: Vince O'Brien
Senior Managing Editor: Scott Disanno
Production Editor: Jane Bonnell
Senior Operations Supervisor: Alan Fischer
Operations Specialist: Lisa McDowell
Marketing Manager: Erin Davis
Marketing Assistant: Mack Patterson
Cover Designers: Kirsten Sims and Geoffrey Cassar
Cover Images: Stan Honda/Getty, Library of Congress, NASA, National Museum of Rome, Peter Norvig, Ian Parker, Shutterstock, Time Life/Getty
Interior Designers: Stuart Russell and Peter Norvig
Copy Editor: Mary Lou Nohr
Art Editor: Greg Dulles
Media Editor: Daniel Sandin
Media Project Manager: Danielle Leone

Library of Congress Cataloging-in-Publication Data on File

Prentice Hall
is an imprint of

PEARSON

www.pearsonhighered.com

10 9 8 7 6 5 4 3 2 1
ISBN-13: 978-0-13-604259-4
ISBN-10: 0-13-604259-7

For Loy, Gordon, Lucy, George, and Isaac — S.J.R.

For Kris, Isabella, and Juliet — P.N.

This page intentionally left blank

Preface

Artificial Intelligence (AI) is a big field, and this is a big book. We have tried to explore the full breadth of the field, which encompasses logic, probability, and continuous mathematics; perception, reasoning, learning, and action; and everything from microelectronic devices to robotic planetary explorers. The book is also big because we go into some depth.

The subtitle of this book is "A Modern Approach." The intended meaning of this rather empty phrase is that we have tried to synthesize what is now known into a common framework, rather than trying to explain each subfield of AI in its own historical context. We apologize to those whose subfields are, as a result, less recognizable.

New to this edition

This edition captures the changes in AI that have taken place since the last edition in 2003. There have been important applications of AI technology, such as the widespread deployment of practical speech recognition, machine translation, autonomous vehicles, and household robotics. There have been algorithmic landmarks, such as the solution of the game of checkers. And there has been a great deal of theoretical progress, particularly in areas such as probabilistic reasoning, machine learning, and computer vision. Most important from our point of view is the continued evolution in how we think about the field, and thus how we organize the book. The major changes are as follows:

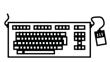
- We place more emphasis on partially observable and nondeterministic environments, especially in the nonprobabilistic settings of search and planning. The concepts of *belief state* (a set of possible worlds) and *state estimation* (maintaining the belief state) are introduced in these settings; later in the book, we add probabilities.
- In addition to discussing the types of environments and types of agents, we now cover in more depth the types of *representations* that an agent can use. We distinguish among *atomic* representations (in which each state of the world is treated as a black box), *factored* representations (in which a state is a set of attribute/value pairs), and *structured* representations (in which the world consists of objects and relations between them).
- Our coverage of planning goes into more depth on contingent planning in partially observable environments and includes a new approach to hierarchical planning.
- We have added new material on first-order probabilistic models, including *open-universe* models for cases where there is uncertainty as to what objects exist.
- We have completely rewritten the introductory machine-learning chapter, stressing a wider variety of more modern learning algorithms and placing them on a firmer theoretical footing.
- We have expanded coverage of Web search and information extraction, and of techniques for learning from very large data sets.
- 20% of the citations in this edition are to works published after 2003.
- We estimate that about 20% of the material is brand new. The remaining 80% reflects older work but has been largely rewritten to present a more unified picture of the field.

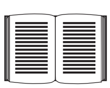
Overview of the book

The main unifying theme is the idea of an **intelligent agent**. We define AI as the study of agents that receive percepts from the environment and perform actions. Each such agent implements a function that maps percept sequences to actions, and we cover different ways to represent these functions, such as reactive agents, real-time planners, and decision-theoretic systems. We explain the role of learning as extending the reach of the designer into unknown environments, and we show how that role constrains agent design, favoring explicit knowledge representation and reasoning. We treat robotics and vision not as independently defined problems, but as occurring in the service of achieving goals. We stress the importance of the task environment in determining the appropriate agent design.

Our primary aim is to convey the *ideas* that have emerged over the past fifty years of AI research and the past two millennia of related work. We have tried to avoid excessive formality in the presentation of these ideas while retaining precision. We have included pseudocode algorithms to make the key ideas concrete; our pseudocode is described in Appendix B.

This book is primarily intended for use in an undergraduate course or course sequence. The book has 27 chapters, each requiring about a week's worth of lectures, so working through the whole book requires a two-semester sequence. A one-semester course can use selected chapters to suit the interests of the instructor and students. The book can also be used in a graduate-level course (perhaps with the addition of some of the primary sources suggested in the bibliographical notes). Sample syllabi are available at the book's Web site, `aima.cs.berkeley.edu`. The only prerequisite is familiarity with basic concepts of computer science (algorithms, data structures, complexity) at a sophomore level. Freshman calculus and linear algebra are useful for some of the topics; the required mathematical background is supplied in Appendix A.





Exercises are given at the end of each chapter. Exercises requiring significant programming are marked with a **keyboard** icon. These exercises can best be solved by taking advantage of the code repository at `aima.cs.berkeley.edu`. Some of them are large enough to be considered term projects. A number of exercises require some investigation of the literature; these are marked with a **book** icon.

Throughout the book, important points are marked with a *pointing* icon. We have included an extensive index of around 6,000 items to make it easy to find things in the book.

NEW TERM

Wherever a **new term** is first defined, it is also marked in the margin.

About the Web site

`aima.cs.berkeley.edu`, the Web site for the book, contains

- implementations of the algorithms in the book in several programming languages,
- a list of over 1000 schools that have used the book, many with links to online course materials and syllabi,
- an annotated list of over 800 links to sites around the Web with useful AI content,
- a chapter-by-chapter list of supplementary material and links,
- instructions on how to join a discussion group for the book,

- instructions on how to contact the authors with questions or comments,
- instructions on how to report errors in the book, in the likely event that some exist, and
- slides and other materials for instructors.

About the cover

The cover depicts the final position from the decisive game 6 of the 1997 match between chess champion Garry Kasparov and program DEEP BLUE. Kasparov, playing Black, was forced to resign, making this the first time a computer had beaten a world champion in a chess match. Kasparov is shown at the top. To his left is the Asimo humanoid robot and to his right is Thomas Bayes (1702–1761), whose ideas about probability as a measure of belief underlie much of modern AI technology. Below that we see a Mars Exploration Rover, a robot that landed on Mars in 2004 and has been exploring the planet ever since. To the right is Alan Turing (1912–1954), whose fundamental work defined the fields of computer science in general and artificial intelligence in particular. At the bottom is Shakey (1966–1972), the first robot to combine perception, world-modeling, planning, and learning. With Shakey is project leader Charles Rosen (1917–2002). At the bottom right is Aristotle (384 B.C.–322 B.C.), who pioneered the study of logic; his work was state of the art until the 19th century (copy of a bust by Lysippos). At the bottom left, lightly screened behind the authors' names, is a planning algorithm by Aristotle from *De Motu Animalium* in the original Greek. Behind the title is a portion of the CPSC Bayesian network for medical diagnosis (Pradhan *et al.*, 1994). Behind the chess board is part of a Bayesian logic model for detecting nuclear explosions from seismic signals.

Credits: Stan Honda/Getty (Kasparaov), Library of Congress (Bayes), NASA (Mars rover), National Museum of Rome (Aristotle), Peter Norvig (book), Ian Parker (Berkeley skyline), Shutterstock (Asimo, Chess pieces), Time Life/Getty (Shakey, Turing).

Acknowledgments

This book would not have been possible without the many contributors whose names did not make it to the cover. Jitendra Malik and David Forsyth wrote Chapter 24 (computer vision) and Sebastian Thrun wrote Chapter 25 (robotics). Vibhu Mittal wrote part of Chapter 22 (natural language). Nick Hay, Mehran Sahami, and Ernest Davis wrote some of the exercises. Zoran Duric (George Mason), Thomas C. Henderson (Utah), Leon Reznik (RIT), Michael Gourley (Central Oklahoma) and Ernest Davis (NYU) reviewed the manuscript and made helpful suggestions. We thank Ernie Davis in particular for his tireless ability to read multiple drafts and help improve the book. Nick Hay whipped the bibliography into shape and on deadline stayed up to 5:30 AM writing code to make the book better. Jon Barron formatted and improved the diagrams in this edition, while Tim Huang, Mark Paskin, and Cynthia Bruyns helped with diagrams and algorithms in previous editions. Ravi Mohan and Ciaran O'Reilly wrote and maintain the Java code examples on the Web site. John Canny wrote the robotics chapter for the first edition and Douglas Edwards researched the historical notes. Tracy Dunkelberger, Allison Michael, Scott Disanno, and Jane Bonnell at Pearson tried their best to keep us on schedule and made many helpful suggestions. Most helpful of all has

been Julie Sussman, P.P.A., who read every chapter and provided extensive improvements. In previous editions we had proofreaders who would tell us when we left out a comma and said *which* when we meant *that*; Julie told us when we left out a minus sign and said x_i when we meant x_j. For every typo or confusing explanation that remains in the book, rest assured that Julie has fixed at least five. She persevered even when a power failure forced her to work by lantern light rather than LCD glow.

Stuart would like to thank his parents for their support and encouragement and his wife, Loy Sheflott, for her endless patience and boundless wisdom. He hopes that Gordon, Lucy, George, and Isaac will soon be reading this book after they have forgiven him for working so long on it. RUGS (Russell's Unusual Group of Students) have been unusually helpful, as always.

Peter would like to thank his parents (Torsten and Gerda) for getting him started, and his wife (Kris), children (Bella and Juliet), colleagues, and friends for encouraging and tolerating him through the long hours of writing and longer hours of rewriting.

We both thank the librarians at Berkeley, Stanford, and NASA and the developers of CiteSeer, Wikipedia, and Google, who have revolutionized the way we do research. We can't acknowledge all the people who have used the book and made suggestions, but we would like to note the especially helpful comments of Gagan Aggarwal, Eyal Amir, Ion Androutsopoulos, Krzysztof Apt, Warren Haley Armstrong, Ellery Aziel, Jeff Van Baalen, Darius Bacon, Brian Baker, Shumeet Baluja, Don Barker, Tony Barrett, James Newton Bass, Don Beal, Howard Beck, Wolfgang Bibel, John Binder, Larry Bookman, David R. Boxall, Ronen Brafman, John Bresina, Gerhard Brewka, Selmer Bringsjord, Carla Brodley, Chris Brown, Emma Brunskill, Wilhelm Burger, Lauren Burka, Carlos Bustamante, Joao Cachopo, Murray Campbell, Norman Carver, Emmanuel Castro, Anil Chakravarthy, Dan Chisarick, Berthe Choueiry, Roberto Cipolla, David Cohen, James Coleman, Julie Ann Comparini, Corinna Cortes, Gary Cottrell, Ernest Davis, Tom Dean, Rina Dechter, Tom Dietterich, Peter Drake, Chuck Dyer, Doug Edwards, Robert Egginton, Asma'a El-Budrawy, Barbara Engelhardt, Kutluhan Erol, Oren Etzioni, Hana Filip, Douglas Fisher, Jeffrey Forbes, Ken Ford, Eric Fosler-Lussier, John Fosler, Jeremy Frank, Alex Franz, Bob Futrelle, Marek Galecki, Stefan Gerberding, Stuart Gill, Sabine Glesner, Seth Golub, Gosta Grahne, Russ Greiner, Eric Grimson, Barbara Grosz, Larry Hall, Steve Hanks, Othar Hansson, Ernst Heinz, Jim Hendler, Christoph Herrmann, Paul Hilfinger, Robert Holte, Vasant Honavar, Tim Huang, Seth Hutchinson, Joost Jacob, Mark Jelasity, Magnus Johansson, Istvan Jonyer, Dan Jurafsky, Leslie Kaelbling, Keiji Kanazawa, Surekha Kasibhatla, Simon Kasif, Henry Kautz, Gernot Kerschbaumer, Max Khesin, Richard Kirby, Dan Klein, Kevin Knight, Roland Koenig, Sven Koenig, Daphne Koller, Rich Korf, Benjamin Kuipers, James Kurien, John Lafferty, John Laird, Gus Larsson, John Lazzaro, Jon LeBlanc, Jason Leatherman, Frank Lee, Jon Lehto, Edward Lim, Phil Long, Pierre Louveaux, Don Loveland, Sridhar Mahadevan, Tony Mancill, Jim Martin, Andy Mayer, John McCarthy, David McGrane, Jay Mendelsohn, Risto Miikkulanien, Brian Milch, Steve Minton, Vibhu Mittal, Mehryar Mohri, Leora Morgenstern, Stephen Muggleton, Kevin Murphy, Ron Musick, Sung Myaeng, Eric Nadeau, Lee Naish, Pandu Nayak, Bernhard Nebel, Stuart Nelson, XuanLong Nguyen, Nils Nilsson, Illah Nourbakhsh, Ali Nouri, Arthur Nunes-Harwitt, Steve Omohundro, David Page, David Palmer, David Parkes, Ron Parr, Mark

Paskin, Tony Passera, Amit Patel, Michael Pazzani, Fernando Pereira, Joseph Perla, Wim Pijls, Ira Pohl, Martha Pollack, David Poole, Bruce Porter, Malcolm Pradhan, Bill Pringle, Lorraine Prior, Greg Provan, William Rapaport, Deepak Ravichandran, Ioannis Refanidis, Philip Resnik, Francesca Rossi, Sam Roweis, Richard Russell, Jonathan Schaeffer, Richard Scherl, Hinrich Schuetze, Lars Schuster, Bart Selman, Soheil Shams, Stuart Shapiro, Jude Shavlik, Yoram Singer, Satinder Singh, Daniel Sleator, David Smith, Bryan So, Robert Sproull, Lynn Stein, Larry Stephens, Andreas Stolcke, Paul Stradling, Devika Subramanian, Marek Suchenek, Rich Sutton, Jonathan Tash, Austin Tate, Bas Terwijn, Olivier Teytaud, Michael Thielscher, William Thompson, Sebastian Thrun, Eric Tiedemann, Mark Torrance, Randall Upham, Paul Utgoff, Peter van Beek, Hal Varian, Paulina Varshavskaya, Sunil Vemuri, Vandi Verma, Ubbo Visser, Jim Waldo, Toby Walsh, Bonnie Webber, Dan Weld, Michael Wellman, Kamin Whitehouse, Michael Dean White, Brian Williams, David Wolfe, Jason Wolfe, Bill Woods, Alden Wright, Jay Yagnik, Mark Yasuda, Richard Yen, Eliezer Yudkowsky, Weixiong Zhang, Ming Zhao, Shlomo Zilberstein, and our esteemed colleague Anonymous Reviewer.

About the Authors

Stuart Russell was born in 1962 in Portsmouth, England. He received his B.A. with first-class honours in physics from Oxford University in 1982, and his Ph.D. in computer science from Stanford in 1986. He then joined the faculty of the University of California at Berkeley, where he is a professor of computer science, director of the Center for Intelligent Systems, and holder of the Smith–Zadeh Chair in Engineering. In 1990, he received the Presidential Young Investigator Award of the National Science Foundation, and in 1995 he was cowinner of the Computers and Thought Award. He was a 1996 Miller Professor of the University of California and was appointed to a Chancellor's Professorship in 2000. In 1998, he gave the Forsythe Memorial Lectures at Stanford University. He is a Fellow and former Executive Council member of the American Association for Artificial Intelligence. He has published over 100 papers on a wide range of topics in artificial intelligence. His other books include *The Use of Knowledge in Analogy and Induction* and (with Eric Wefald) *Do the Right Thing: Studies in Limited Rationality.*

Peter Norvig is currently Director of Research at Google, Inc., and was the director responsible for the core Web search algorithms from 2002 to 2005. He is a Fellow of the American Association for Artificial Intelligence and the Association for Computing Machinery. Previously, he was head of the Computational Sciences Division at NASA Ames Research Center, where he oversaw NASA's research and development in artificial intelligence and robotics, and chief scientist at Junglee, where he helped develop one of the first Internet information extraction services. He received a B.S. in applied mathematics from Brown University and a Ph.D. in computer science from the University of California at Berkeley. He received the Distinguished Alumni and Engineering Innovation awards from Berkeley and the Exceptional Achievement Medal from NASA. He has been a professor at the University of Southern California and a research faculty member at Berkeley. His other books are *Paradigms of AI Programming: Case Studies in Common Lisp* and *Verbmobil: A Translation System for Face-to-Face Dialog* and *Intelligent Help Systems for UNIX*.

Contents

for belief-state search; these were refined by Bryce *et al.* (2006). The incremental approach to belief-state search, in which solutions are constructed incrementally for subsets of states within each belief state, was studied in the planning literature by Kurien *et al.* (2002); several new incremental algorithms were introduced for nondeterministic, partially observable problems by Russell and Wolfe (2005). Additional references for planning in stochastic, partially observable environments appear in Chapter 17.

EULERIAN GRAPH

Algorithms for exploring unknown state spaces have been of interest for many centuries. Depth-first search in a maze can be implemented by keeping one's left hand on the wall; loops can be avoided by marking each junction. Depth-first search fails with irreversible actions; the more general problem of exploring **Eulerian graphs** (i.e., graphs in which each node has equal numbers of incoming and outgoing edges) was solved by an algorithm due to Hierholzer (1873). The first thorough algorithmic study of the exploration problem for arbitrary graphs was carried out by Deng and Papadimitriou (1990), who developed a completely general algorithm but showed that no bounded competitive ratio is possible for exploring a general graph. Papadimitriou and Yannakakis (1991) examined the question of finding paths to a goal in geometric path-planning environments (where all actions are reversible). They showed that a small competitive ratio is achievable with square obstacles, but with general rectangular obstacles no bounded ratio can be achieved. (See Figure 4.20.)

REAL-TIME SEARCH

The LRTA* algorithm was developed by Korf (1990) as part of an investigation into **real-time search** for environments in which the agent must act after searching for only a fixed amount of time (a common situation in two-player games). LRTA* is in fact a special case of reinforcement learning algorithms for stochastic environments (Barto *et al.*, 1995). Its policy of optimism under uncertainty—always head for the closest unvisited state—can result in an exploration pattern that is less efficient in the uninformed case than simple depth-first search (Koenig, 2000). Dasgupta *et al.* (1994) show that online iterative deepening search is optimally efficient for finding a goal in a uniform tree with no heuristic information. Several informed variants on the LRTA* theme have been developed with different methods for searching and updating within the known portion of the graph (Pemberton and Korf, 1992). As yet, there is no good understanding of how to find goals with optimal efficiency when using heuristic information.

EXERCISES

4.1 Give the name of the algorithm that results from each of the following special cases:

a. Local beam search with $k = 1$.

b. Local beam search with one initial state and no limit on the number of states retained.

c. Simulated annealing with $T = 0$ at all times (and omitting the termination test).

d. Simulated annealing with $T = \infty$ at all times.

e. Genetic algorithm with population size $N = 1$.

4.2 Exercise 3.16 considers the problem of building railway tracks under the assumption that pieces fit exactly with no slack. Now consider the real problem, in which pieces don't fit exactly but allow for up to 10 degrees of rotation to either side of the "proper" alignment. Explain how to formulate the problem so it could be solved by simulated annealing.

4.3 In this exercise, we explore the use of local search methods to solve TSPs of the type defined in Exercise 3.30.

a. Implement and test a hill-climbing method to solve TSPs. Compare the results with optimal solutions obtained from the A* algorithm with the MST heuristic (Exercise 3.30).

b. Repeat part (a) using a genetic algorithm instead of hill climbing. You may want to consult Larrañaga *et al.* (1999) for some suggestions for representations.

4.4 Generate a large number of 8-puzzle and 8-queens instances and solve them (where possible) by hill climbing (steepest-ascent and first-choice variants), hill climbing with random restart, and simulated annealing. Measure the search cost and percentage of solved problems and graph these against the optimal solution cost. Comment on your results.

4.5 The AND-OR-GRAPH-SEARCH algorithm in Figure 4.11 checks for repeated states only on the path from the root to the current state. Suppose that, in addition, the algorithm were to store *every* visited state and check against that list. (See BREADTH-FIRST-SEARCH in Figure 3.11 for an example.) Determine the information that should be stored and how the algorithm should use that information when a repeated state is found. (*Hint*: You will need to distinguish at least between states for which a successful subplan was constructed previously and states for which no subplan could be found.) Explain how to use labels, as defined in Section 4.3.3, to avoid having multiple copies of subplans.

4.6 Explain precisely how to modify the AND-OR-GRAPH-SEARCH algorithm to generate a cyclic plan if no acyclic plan exists. You will need to deal with three issues: labeling the plan steps so that a cyclic plan can point back to an earlier part of the plan, modifying OR-SEARCH so that it continues to look for acyclic plans after finding a cyclic plan, and augmenting the plan representation to indicate whether a plan is cyclic. Show how your algorithm works on (a) the slippery vacuum world, and (b) the slippery, erratic vacuum world. You might wish to use a computer implementation to check your results.

4.7 In Section 4.4.1 we introduced belief states to solve sensorless search problems. A sequence of actions solves a sensorless problem if it maps every physical state in the initial belief state b to a goal state. Suppose the agent knows $h^*(s)$, the true optimal cost of solving the physical state s in the fully observable problem, for every state s in b. Find an admissible heuristic $h(b)$ for the sensorless problem in terms of these costs, and prove its admissibilty. Comment on the accuracy of this heuristic on the sensorless vacuum problem of Figure 4.14. How well does A* perform?

4.8 This exercise explores subset–superset relations between belief states in sensorless or partially observable environments.

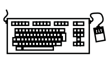
a. Prove that if an action sequence is a solution for a belief state b, it is also a solution for any subset of b. Can anything be said about supersets of b?

b. Explain in detail how to modify graph search for sensorless problems to take advantage of your answers in (a).

c. Explain in detail how to modify AND–OR search for partially observable problems, beyond the modifications you describe in (b).

4.9 On page 139 it was assumed that a given action would have the same cost when executed in any physical state within a given belief state. (This leads to a belief-state search problem with well-defined step costs.) Now consider what happens when the assumption does not hold. Does the notion of optimality still make sense in this context, or does it require modification? Consider also various possible definitions of the "cost" of executing an action in a belief state; for example, we could use the *minimum* of the physical costs; or the *maximum*; or a cost *interval* with the lower bound being the minimum cost and the upper bound being the maximum; or just keep the set of all possible costs for that action. For each of these, explore whether A* (with modifications if necessary) can return optimal solutions.

4.10 Consider the sensorless version of the erratic vacuum world. Draw the belief-state space reachable from the initial belief state $\{1, 2, 3, 4, 5, 6, 7, 8\}$, and explain why the problem is unsolvable.



4.11 We can turn the navigation problem in Exercise 3.7 into an environment as follows:

- The percept will be a list of the positions, *relative to the agent*, of the visible vertices. The percept does *not* include the position of the robot! The robot must learn its own position from the map; for now, you can assume that each location has a different "view."
- Each action will be a vector describing a straight-line path to follow. If the path is unobstructed, the action succeeds; otherwise, the robot stops at the point where its path first intersects an obstacle. If the agent returns a zero motion vector and is at the goal (which is fixed and known), then the environment teleports the agent to a *random location* (not inside an obstacle).
- The performance measure charges the agent 1 point for each unit of distance traversed and awards 1000 points each time the goal is reached.

a. Implement this environment and a problem-solving agent for it. After each teleportation, the agent will need to formulate a new problem, which will involve discovering its current location.

b. Document your agent's performance (by having the agent generate suitable commentary as it moves around) and report its performance over 100 episodes.

c. Modify the environment so that 30% of the time the agent ends up at an unintended destination (chosen randomly from the other visible vertices if any; otherwise, no move at all). This is a crude model of the motion errors of a real robot. Modify the agent so that when such an error is detected, it finds out where it is and then constructs a plan to get back to where it was and resume the old plan. Remember that sometimes getting back to where it was might also fail! Show an example of the agent successfully overcoming two successive motion errors and still reaching the goal.

d. Now try two different recovery schemes after an error: (1) head for the closest vertex on the original route; and (2) replan a route to the goal from the new location. Compare the performance of the three recovery schemes. Would the inclusion of search costs affect the comparison?

e. Now suppose that there are locations from which the view is identical. (For example, suppose the world is a grid with square obstacles.) What kind of problem does the agent now face? What do solutions look like?

4.12 Suppose that an agent is in a 3×3 maze environment like the one shown in Figure 4.19. The agent knows that its initial location is (1,1), that the goal is at (3,3), and that the actions Up, $Down$, $Left$, $Right$ have their usual effects unless blocked by a wall. The agent does *not* know where the internal walls are. In any given state, the agent perceives the set of legal actions; it can also tell whether the state is one it has visited before.

a. Explain how this online search problem can be viewed as an offline search in belief-state space, where the initial belief state includes all possible environment configurations. How large is the initial belief state? How large is the space of belief states?

b. How many distinct percepts are possible in the initial state?

c. Describe the first few branches of a contingency plan for this problem. How large (roughly) is the complete plan?

Notice that this contingency plan is a solution for *every possible environment* fitting the given description. Therefore, interleaving of search and execution is not strictly necessary even in unknown environments.

4.13 In this exercise, we examine hill climbing in the context of robot navigation, using the environment in Figure 3.31 as an example.

a. Repeat Exercise 4.11 using hill climbing. Does your agent ever get stuck in a local minimum? Is it *possible* for it to get stuck with convex obstacles?

b. Construct a nonconvex polygonal environment in which the agent gets stuck.

c. Modify the hill-climbing algorithm so that, instead of doing a depth-1 search to decide where to go next, it does a depth-k search. It should find the best k-step path and do one step along it, and then repeat the process.

d. Is there some k for which the new algorithm is guaranteed to escape from local minima?

e. Explain how LRTA* enables the agent to escape from local minima in this case.

4.14 Like DFS, online DFS is incomplete for reversible state spaces with infinite paths. For example, suppose that states are points on the infinite two-dimensional grid and actions are unit vectors $(1,0)$, $(0,1)$, $(-1,0)$, $(0,-1)$, tried in that order. Show that online DFS starting at $(0,0)$ will not reach $(1,-1)$. Suppose the agent can observe, in addition to its current state, all successor states and the actions that would lead to them. Write an algorithm that is complete even for bidirected state spaces with infinite paths. What states does it visit in reaching $(1,-1)$?