[Artificial Intelligence: A Modern Approach, 4th US ed.](http://aima.cs.berkeley.edu/)

[aimacode · GitHub](https://github.com/aimacode)

**1. Introduction**

These exercises are intended to stimulate discussion, and some might be set as term projects. Alternatively, preliminary attempts can be made now, and these attempts can be reviewed after the completion of the book.

[Exercise 1](https://aimacode.github.io/aima-exercises/intro-exercises/ex_1/)

Define in your own words: (a) intelligence, (b) artificial intelligence, (c) agent, (d) rationality, (e) logical reasoning.

[Exercise 2](https://aimacode.github.io/aima-exercises/intro-exercises/ex_2/)

Read Turing’s original paper on AI [Turing:1950](https://aimacode.github.io/aima-exercises/intro-exercises/).In the paper, he discusses several objections to his proposed enterprise and his test for intelligence. Which objections still carry weight? Are his refutations valid? Can you think of new objections arising from developments since he wrote the paper? In the paper, he predicts that, by the year 2000, a computer will have a 30% chance of passing a five-minute Turing Test with an unskilled interrogator. What chance do you think a computer would have today? In another 50 years?

[Exercise 3](https://aimacode.github.io/aima-exercises/intro-exercises/ex_3/)

Every year the Loebner Prize is awarded to the program that comes closest to passing a version of the [Turing Test](https://en.wikipedia.org/wiki/Turing_test). Research and report on the latest winner of the Loebner prize. What techniques does it use? How does it advance the state of the art in AI?

[Exercise 4](https://aimacode.github.io/aima-exercises/intro-exercises/ex_4/)

Are reflex actions (such as flinching from a hot stove) rational? Are they intelligent?

[Exercise 5](https://aimacode.github.io/aima-exercises/intro-exercises/ex_5/)

There are well-known classes of problems that are intractably difficult for computers, and other classes that are provably undecidable. Does this mean that AI is impossible?

[Exercise 6](https://aimacode.github.io/aima-exercises/intro-exercises/ex_6/)

Suppose we extend Evans’s *SYSTEM* program so that it can score 200 on a standard IQ test. Would we then have a program more intelligent than a human? Explain.

[Exercise 7](https://aimacode.github.io/aima-exercises/intro-exercises/ex_7/)

The neural structure of the sea slug *Aplysis* has been widely studied (first by Nobel Laureate Eric Kandel) because it has only about 20,000 neurons, most of them large and easily manipulated. Assuming that the cycle time for an *Aplysis* neuron is roughly the same as for a human neuron, how does the computational power, in terms of memory updates per second, compare with the high-end computer described in (Figure [1.3](https://aimacode.github.io/aima-exercises/figures/computer-brain-table.png))?

[Exercise 8](https://aimacode.github.io/aima-exercises/intro-exercises/ex_8/)

How could introspection—reporting on one’s inner thoughts—be inaccurate? Could I be wrong about what I’m thinking? Discuss.

[Exercise 9](https://aimacode.github.io/aima-exercises/intro-exercises/ex_9/)

To what extent are the following computer systems instances of artificial intelligence:  
- Supermarket bar code scanners.  
- Web search engines.  
- Voice-activated telephone menus.  
- Internet routing algorithms that respond dynamically to the state of the network.

[Exercise 10](https://aimacode.github.io/aima-exercises/intro-exercises/ex_10/)

To what extent are the following computer systems instances of artificial intelligence:  
- Supermarket bar code scanners.  
- Voice-activated telephone menus.  
- Spelling and grammar correction features in Microsoft Word.  
- Internet routing algorithms that respond dynamically to the state of the network.

[Exercise 11](https://aimacode.github.io/aima-exercises/intro-exercises/ex_11/)

Many of the computational models of cognitive activities that have been proposed involve quite complex mathematical operations, such as convolving an image with a Gaussian or finding a minimum of the entropy function. Most humans (and certainly all animals) never learn this kind of mathematics at all, almost no one learns it before college, and almost no one can compute the convolution of a function with a Gaussian in their head. What sense does it make to say that the “vision system” is doing this kind of mathematics, whereas the actual person has no idea how to do it?

[Exercise 12](https://aimacode.github.io/aima-exercises/intro-exercises/ex_12/)

Some authors have claimed that perception and motor skills are the most important part of intelligence, and that “higher level” capacities are necessarily parasitic—simple add-ons to these underlying facilities. Certainly, most of evolution and a large part of the brain have been devoted to perception and motor skills, whereas AI has found tasks such as game playing and logical inference to be easier, in many ways, than perceiving and acting in the real world. Do you think that AI’s traditional focus on higher-level cognitive abilities is misplaced?

[Exercise 13](https://aimacode.github.io/aima-exercises/intro-exercises/ex_13/)

Why would evolution tend to result in systems that act rationally? What goals are such systems designed to achieve?

[Exercise 14](https://aimacode.github.io/aima-exercises/intro-exercises/ex_14/)

Is AI a science, or is it engineering? Or neither or both? Explain.

[Exercise 15](https://aimacode.github.io/aima-exercises/intro-exercises/ex_15/)

“Surely computers cannot be intelligent—they can do only what their programmers tell them.” Is the latter statement true, and does it imply the former?

[Exercise 16](https://aimacode.github.io/aima-exercises/intro-exercises/ex_16/)

“Surely animals cannot be intelligent—they can do only what their genes tell them.” Is the latter statement true, and does it imply the former?

[Exercise 17](https://aimacode.github.io/aima-exercises/intro-exercises/ex_17/)

“Surely animals, humans, and computers cannot be intelligent—they can do only what their constituent atoms are told to do by the laws of physics.” Is the latter statement true, and does it imply the former?

[Exercise 18](https://aimacode.github.io/aima-exercises/intro-exercises/ex_18/)

Examine the AI literature to discover whether the following tasks can currently be solved by computers: - Playing a decent game of table tennis (Ping-Pong). - Driving in the center of Cairo, Egypt. - Driving in Victorville, California. - Buying a week’s worth of groceries at the market. - Buying a week’s worth of groceries on the Web. - Playing a decent game of bridge at a competitive level. - Discovering and proving new mathematical theorems. - Writing an intentionally funny story. - Giving competent legal advice in a specialized area of law. - Translating spoken English into spoken Swedish in real time. - Performing a complex surgical operation.

[Exercise 19](https://aimacode.github.io/aima-exercises/intro-exercises/ex_19/)

For the currently infeasible tasks, try to find out what the difficulties are and predict when, if ever, they will be overcome.

[Exercise 20](https://aimacode.github.io/aima-exercises/intro-exercises/ex_20/)

Various subfields of AI have held contests by defining a standard task and inviting researchers to do their best. Examples include the DARPA Grand Challenge for robotic cars, the International Planning Competition, the Robocup robotic soccer league, the TREC information retrieval event, and contests in machine translation and speech recognition. Investigate five of these contests and describe the progress made over the years. To what degree have the contests advanced the state of the art in AI? To what degree do they hurt the field by drawing energy away from new ideas?

**2. Intelligent Agents**

[Exercise 1](https://aimacode.github.io/aima-exercises/agents-exercises/ex_1/)

Suppose that the performance measure is concerned with just the first TT time steps of the environment and ignores everything thereafter. Show that a rational agent’s action may depend not just on the state of the environment but also on the time step it has reached.

[Exercise 2 (vacuum-rationality-exercise)](https://aimacode.github.io/aima-exercises/agents-exercises/ex_2/)

Let us examine the rationality of various vacuum-cleaner agent functions.  
1. Show that the simple vacuum-cleaner agent function described in Figure [2.3](https://aimacode.github.io/aima-exercises/figures/vacuum-agent-function-table.png) is indeed rational under the assumptions listed on page   
2. Describe a rational agent function for the case in which each movement costs one point. Does the corresponding agent program require internal state?  
3. Discuss possible agent designs for the cases in which clean squares can become dirty and the geography of the environment is unknown. Does it make sense for the agent to learn from its experience in these cases? If so, what should it learn? If not, why not?

[Exercise 3](https://aimacode.github.io/aima-exercises/agents-exercises/ex_3/)

Write an essay on the relationship between evolution and one or more of autonomy, intelligence, and learning.

[Exercise 4](https://aimacode.github.io/aima-exercises/agents-exercises/ex_4/)

For each of the following assertions, say whether it is true or false and support your answer with examples or counterexamples where appropriate.  
1. An agent that senses only partial information about the state cannot be perfectly rational.  
2. There exist task environments in which no pure reflex agent can behave rationally.  
3. There exists a task environment in which every agent is rational.  
4. The input to an agent program is the same as the input to the agent function.  
5. Every agent function is implementable by some program/machine combination.  
6. Suppose an agent selects its action uniformly at random from the set of possible actions. There exists a deterministic task environment in which this agent is rational.  
7. It is possible for a given agent to be perfectly rational in two distinct task environments.  
8. Every agent is rational in an unobservable environment.  
9. A perfectly rational poker-playing agent never loses.

[Exercise 5 (PEAS-exercise)](https://aimacode.github.io/aima-exercises/agents-exercises/ex_5/)

For each of the following activities, give a PEAS description of the task environment and characterize it in terms of the properties listed in Section   
- Playing soccer.  
- Exploring the subsurface oceans of Titan.  
- Shopping for used AI books on the Internet.  
- Playing a tennis match.  
- Practicing tennis against a wall.  
- Performing a high jump.  
- Knitting a sweater.  
- Bidding on an item at an auction.

[Exercise 6](https://aimacode.github.io/aima-exercises/agents-exercises/ex_6/)

For each of the following activities, give a PEAS description of the task environment and characterize it in terms of the properties listed in Section   
- Performing a gymnastics floor routine.  
- Exploring the subsurface oceans of Titan.  
- Playing soccer.  
- Shopping for used AI books on the Internet.  
- Practicing tennis against a wall.  
- Performing a high jump.  
- Bidding on an item at an auction.

[Exercise 7 (agent-fn-prog-exercise)](https://aimacode.github.io/aima-exercises/agents-exercises/ex_7/)

Define in your own words the following terms: agent, agent function, agent program, rationality, autonomy, reflex agent, model-based agent, goal-based agent, utility-based agent, learning agent.

[Exercise 8](https://aimacode.github.io/aima-exercises/agents-exercises/ex_8/)

This exercise explores the differences between agent functions and agent programs.  
1. Can there be more than one agent program that implements a given agent function? Give an example, or show why one is not possible.  
2. Are there agent functions that cannot be implemented by any agent program?  
3. Given a fixed machine architecture, does each agent program implement exactly one agent function?  
4. Given an architecture with nn bits of storage, how many different possible agent programs are there?  
5. Suppose we keep the agent program fixed but speed up the machine by a factor of two. Does that change the agent function?

[Exercise 9](https://aimacode.github.io/aima-exercises/agents-exercises/ex_9/)

Write pseudocode agent programs for the goal-based and utility-based agents.

The following exercises all concern the implementation of environments and agents for the vacuum-cleaner world.

[Exercise 10 (vacuum-start-exercise)](https://aimacode.github.io/aima-exercises/agents-exercises/ex_10/)

Consider a simple thermostat that turns on a furnace when the temperature is at least 3 degrees below the setting, and turns off a furnace when the temperature is at least 3 degrees above the setting. Is a thermostat an instance of a simple reflex agent, a model-based reflex agent, or a goal-based agent?

[Exercise 11](https://aimacode.github.io/aima-exercises/agents-exercises/ex_11/)

Implement a performance-measuring environment simulator for the vacuum-cleaner world depicted in Figure [2.8](https://aimacode.github.io/aima-exercises/figures/vacuum-world-figure.png) and specified on page . Your implementation should be modular so that the sensors, actuators, and environment characteristics (size, shape, dirt placement, etc.) can be changed easily. (Note: for some choices of programming language and operating system there are already implementations in the online code repository.)

[Exercise 12 (vacuum-motion-penalty-exercise)](https://aimacode.github.io/aima-exercises/agents-exercises/ex_12/)

Implement a simple reflex agent for the vacuum environment in Exercise [2.10](https://aimacode.github.io/aima-exercises/agents-exercises/ex_10). Run the environment with this agent for all possible initial dirt configurations and agent locations. Record the performance score for each configuration and the overall average score.

[Exercise 13 (vacuum-unknown-geog-exercise)](https://aimacode.github.io/aima-exercises/agents-exercises/ex_13/)

Consider a modified version of the vacuum environment in Exercise [2.10](https://aimacode.github.io/aima-exercises/agents-exercises/ex_10), in which the agent is penalized one point for each movement.  
1. Can a simple reflex agent be perfectly rational for this environment? Explain.  
2. What about a reflex agent with state? Design such an agent.  
3. How do your answers to 1 and 2 change if the agent’s percepts give it the clean/dirty status of every square in the environment?

[Exercise 14 (vacuum-bump-exercise)](https://aimacode.github.io/aima-exercises/agents-exercises/ex_14/)

Consider a modified version of the vacuum environment in Exercise [2.10](https://aimacode.github.io/aima-exercises/agents-exercises/ex_10), in which the geography of the environment—its extent, boundaries, and obstacles—is unknown, as is the initial dirt configuration. (The agent can go Up and Down as well as Left and Right.)  
1. Can a simple reflex agent be perfectly rational for this environment? Explain.  
2. Can a simple reflex agent with a randomized agent function outperform a simple reflex agent? Design such an agent and measure its performance on several environments.  
3. Can you design an environment in which your randomized agent will perform poorly? Show your results.  
4. Can a reflex agent with state outperform a simple reflex agent? Design such an agent and measure its performance on several environments. Can you design a rational agent of this type?

[Exercise 15 (vacuum-finish-exercise)](https://aimacode.github.io/aima-exercises/agents-exercises/ex_15/)

Repeat Exercise [2.13](https://aimacode.github.io/aima-exercises/agents-exercises/ex_13) for the case in which the location sensor is replaced with a “bump” sensor that detects the agent’s attempts to move into an obstacle or to cross the boundaries of the environment. Suppose the bump sensor stops working; how should the agent behave?

[Exercise 16](https://aimacode.github.io/aima-exercises/agents-exercises/ex_15/)

The vacuum environments in the preceding exercises have all been deterministic. Discuss possible agent programs for each of the following stochastic versions:  
1. Murphy’s law: twenty-five percent of the time, the Suck action fails to clean the floor if it is dirty and deposits dirt onto the floor if the floor is clean. How is your agent program affected if the dirt sensor gives the wrong answer 10% of the time?  
2. Small children: At each time step, each clean square has a 10% chance of becoming dirty. Can you come up with a rational agent design for this case?

**3. Solving Problems By Searching**

[Exercise 1](https://aimacode.github.io/aima-exercises/search-exercises/ex_1/)

Explain why problem formulation must follow goal formulation.

[Exercise 2](https://aimacode.github.io/aima-exercises/search-exercises/ex_2/)

Give a complete problem formulation for each of the following problems. Choose a formulation that is precise enough to be implemented.  
1. There are six glass boxes in a row, each with a lock. Each of the first five boxes holds a key unlocking the next box in line; the last box holds a banana. You have the key to the first box, and you want the banana.  
2. You start with the sequence ABABAECCEC, or in general any sequence made from A, B, C, and E. You can transform this sequence using the following equalities: AC = E, AB = BC, BB = E, and Exx = xx for any xx. For example, ABBC can be transformed into AEC, and then AC, and then E. Your goal is to produce the sequence E.  
3. There is an n×nn×n grid of squares, each square initially being either unpainted floor or a bottomless pit. You start standing on an unpainted floor square, and can either paint the square under you or move onto an adjacent unpainted floor square. You want the whole floor painted.  
4. A container ship is in port, loaded high with containers. There 13 rows of containers, each 13 containers wide and 5 containers tall. You control a crane that can move to any location above the ship, pick up the container under it, and move it onto the dock. You want the ship unloaded.

[Exercise 3](https://aimacode.github.io/aima-exercises/search-exercises/ex_3/)

Your goal is to navigate a robot out of a maze. The robot starts in the center of the maze facing north. You can turn the robot to face north, east, south, or west. You can direct the robot to move forward a certain distance, although it will stop before hitting a wall.  
1. Formulate this problem. How large is the state space?  
2. In navigating a maze, the only place we need to turn is at the intersection of two or more corridors. Reformulate this problem using this observation. How large is the state space now?  
3. From each point in the maze, we can move in any of the four directions until we reach a turning point, and this is the only action we need to do. Reformulate the problem using these actions. Do we need to keep track of the robot’s orientation now?  
4. In our initial description of the problem we already abstracted from the real world, restricting actions and removing details. List three such simplifications we made.

[Exercise 4](https://aimacode.github.io/aima-exercises/search-exercises/ex_4/)

You have a 9×99×9 grid of squares, each of which can be colored red or blue. The grid is initially colored all blue, but you can change the color of any square any number of times. Imagining the grid divided into nine 3×33×3 sub-squares, you want each sub-square to be all one color but neighboring sub-squares to be different colors.  
1. Formulate this problem in the straightforward way. Compute the size of the state space.  
2. You need color a square only once. Reformulate, and compute the size of the state space. Would breadth-first graph search perform faster on this problem than on the one in (a)? How about iterative deepening tree search?  
3. Given the goal, we need consider only colorings where each sub-square is uniformly colored. Reformulate the problem and compute the size of the state space.  
4. How many solutions does this problem have?  
5. Parts (b) and (c) successively abstracted the original problem (a). Can you give a translation from solutions in problem (c) into solutions in problem (b), and from solutions in problem (b) into solutions for problem (a)?

[Exercise 5 (two-friends-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_5/)

Suppose two friends live in different cities on a map, such as the Romania map shown in . On every turn, we can simultaneously move each friend to a neighboring city on the map. The amount of time needed to move from city ii to neighbor jj is equal to the road distance d(i,j)d(i,j) between the cities, but on each turn the friend that arrives first must wait until the other one arrives (and calls the first on his/her cell phone) before the next turn can begin. We want the two friends to meet as quickly as possible.  
1. Write a detailed formulation for this search problem. (You will find it helpful to define some formal notation here.)  
2. Let D(i,j)D(i,j) be the straight-line distance between cities ii and jj. Which of the following heuristic functions are admissible? (i) D(i,j)D(i,j); (ii) 2⋅D(i,j)2⋅D(i,j); (iii) D(i,j)/2D(i,j)/2.  
3. Are there completely connected maps for which no solution exists?  
4. Are there maps in which all solutions require one friend to visit the same city twice?

[Exercise 6 (8puzzle-parity-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_6/)

Show that the 8-puzzle states are divided into two disjoint sets, such that any state is reachable from any other state in the same set, while no state is reachable from any state in the other set. (*Hint:* See [Berlekamp+al:1982](https://aimacode.github.io/aima-exercises/search-exercises/)) Devise a procedure to decide which set a given state is in, and explain why this is useful for generating random states.

[Exercise 7 (nqueens-size-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_7/)

Consider the nn-queens problem using the “efficient” incremental formulation given on page . Explain why the state space has at least n!−−√3n!3 states and estimate the largest nn for which exhaustive exploration is feasible. (*Hint*: Derive a lower bound on the branching factor by considering the maximum number of squares that a queen can attack in any column.)

[Exercise 8](https://aimacode.github.io/aima-exercises/search-exercises/ex_8/)

Give a complete problem formulation for each of the following. Choose a formulation that is precise enough to be implemented.  
1. Using only four colors, you have to color a planar map in such a way that no two adjacent regions have the same color.  
2. A 3-foot-tall monkey is in a room where some bananas are suspended from the 8-foot ceiling. He would like to get the bananas. The room contains two stackable, movable, climbable 3-foot-high crates.  
3. You have a program that outputs the message “illegal input record” when fed a certain file of input records. You know that processing of each record is independent of the other records. You want to discover what record is illegal.  
4. You have three jugs, measuring 12 gallons, 8 gallons, and 3 gallons, and a water faucet. You can fill the jugs up or empty them out from one to another or onto the ground. You need to measure out exactly one gallon.

[Exercise 9 (path-planning-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_9/)

Consider the problem of finding the shortest path between two points on a plane that has convex polygonal obstacles as shown in . This is an idealization of the problem that a robot has to solve to navigate in a crowded environment.  
1. Suppose the state space consists of all positions (x,y)(x,y) in the plane. How many states are there? How many paths are there to the goal?  
2. Explain briefly why the shortest path from one polygon vertex to any other in the scene must consist of straight-line segments joining some of the vertices of the polygons. Define a good state space now. How large is this state space?  
3. Define the necessary functions to implement the search problem, including an function that takes a vertex as input and returns a set of vectors, each of which maps the current vertex to one of the vertices that can be reached in a straight line. (Do not forget the neighbors on the same polygon.) Use the straight-line distance for the heuristic function.  
4. Apply one or more of the algorithms in this chapter to solve a range of problems in the domain, and comment on their performance.

[Exercise 10 (negative-g-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_10/)

On page , we said that we would not consider problems with negative path costs. In this exercise, we explore this decision in more depth.  
1. Suppose that actions can have arbitrarily large negative costs; explain why this possibility would force any optimal algorithm to explore the entire state space.  
2. Does it help if we insist that step costs must be greater than or equal to some negative constant cc? Consider both trees and graphs.  
3. Suppose that a set of actions forms a loop in the state space such that executing the set in some order results in no net change to the state. If all of these actions have negative cost, what does this imply about the optimal behavior for an agent in such an environment?  
4. One can easily imagine actions with high negative cost, even in domains such as route finding. For example, some stretches of road might have such beautiful scenery as to far outweigh the normal costs in terms of time and fuel. Explain, in precise terms, within the context of state-space search, why humans do not drive around scenic loops indefinitely, and explain how to define the state space and actions for route finding so that artificial agents can also avoid looping.  
5. Can you think of a real domain in which step costs are such as to cause looping?

[Exercise 11 (mc-problem)](https://aimacode.github.io/aima-exercises/search-exercises/ex_11/)

The problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place. This problem is famous in AI because it was the subject of the first paper that approached problem formulation from an analytical viewpoint [Amarel:1968](https://aimacode.github.io/aima-exercises/search-exercises/).  
1. Formulate the problem precisely, making only those distinctions necessary to ensure a valid solution. Draw a diagram of the complete state space.  
2. Implement and solve the problem optimally using an appropriate search algorithm. Is it a good idea to check for repeated states?  
3. Why do you think people have a hard time solving this puzzle, given that the state space is so simple?

[Exercise 12](https://aimacode.github.io/aima-exercises/search-exercises/ex_1/)

Define in your own words the following terms: state, state space, search tree, search node, goal, action, transition model, and branching factor.

[Exercise 13](https://aimacode.github.io/aima-exercises/search-exercises/ex_13/)

What’s the difference between a world state, a state description, and a search node? Why is this distinction useful?

[Exercise 14](https://aimacode.github.io/aima-exercises/search-exercises/ex_14/)

An action such as really consists of a long sequence of finer-grained actions: turn on the car, release the brake, accelerate forward, etc. Having composite actions of this kind reduces the number of steps in a solution sequence, thereby reducing the search time. Suppose we take this to the logical extreme, by making super-composite actions out of every possible sequence of actions. Then every problem instance is solved by a single super-composite action, such as . Explain how search would work in this formulation. Is this a practical approach for speeding up problem solving?

[Exercise 15](https://aimacode.github.io/aima-exercises/search-exercises/ex_15/)

Does a finite state space always lead to a finite search tree? How about a finite state space that is a tree? Can you be more precise about what types of state spaces always lead to finite search trees? (Adapted from , 1996.)

[Exercise 16 (graph-separation-property-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_16/)

Prove that satisfies the graph separation property illustrated in . (*Hint*: Begin by showing that the property holds at the start, then show that if it holds before an iteration of the algorithm, it holds afterwards.) Describe a search algorithm that violates the property.

[Exercise 17](https://aimacode.github.io/aima-exercises/search-exercises/ex_17/)

Which of the following are true and which are false? Explain your answers.  
1. Depth-first search always expands at least as many nodes as A search with an admissible heuristic.  
2. h(n)=0h(n)=0 is an admissible heuristic for the 8-puzzle.  
3. A is of no use in robotics because percepts, states, and actions are continuous.  
4. Breadth-first search is complete even if zero step costs are allowed.  
5. Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.

[Exercise 18](https://aimacode.github.io/aima-exercises/search-exercises/ex_18/)

Consider a state space where the start state is number 1 and each state kk has two successors: numbers 2k2k and 2k+12k+1.  
1. Draw the portion of the state space for states 1 to 15.  
2. Suppose the goal state is 11. List the order in which nodes will be visited for breadth-first search, depth-limited search with limit 3, and iterative deepening search.  
3. How well would bidirectional search work on this problem? What is the branching factor in each direction of the bidirectional search?  
4. Does the answer to (c) suggest a reformulation of the problem that would allow you to solve the problem of getting from state 1 to a given goal state with almost no search?  
5. Call the action going from kk to 2k2k Left, and the action going to 2k+12k+1 Right. Can you find an algorithm that outputs the solution to this problem without any search at all?

[Exercise 19 (brio-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_19/)

A basic wooden railway set contains the pieces shown in . The task is to connect these pieces into a railway that has no overlapping tracks and no loose ends where a train could run off onto the floor.  
1. Suppose that the pieces fit together *exactly* with no slack. Give a precise formulation of the task as a search problem.  
2. Identify a suitable uninformed search algorithm for this task and explain your choice.  
3. Explain why removing any one of the “fork” pieces makes the problem unsolvable.  
4. Give an upper bound on the total size of the state space defined by your formulation. (*Hint*: think about the maximum branching factor for the construction process and the maximum depth, ignoring the problem of overlapping pieces and loose ends. Begin by pretending that every piece is unique.)

[Exercise 20](https://aimacode.github.io/aima-exercises/search-exercises/ex_20/)

Implement two versions of the function for the 8-puzzle: one that copies and edits the data structure for the parent node ss and one that modifies the parent state directly (undoing the modifications as needed). Write versions of iterative deepening depth-first search that use these functions and compare their performance.

[Exercise 21 (iterative-lengthening-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_21/)

On page , we mentioned **iterative lengthening search**, an iterative analog of uniform cost search. The idea is to use increasing limits on path cost. If a node is generated whose path cost exceeds the current limit, it is immediately discarded. For each new iteration, the limit is set to the lowest path cost of any node discarded in the previous iteration.  
1. Show that this algorithm is optimal for general path costs.  
2. Consider a uniform tree with branching factor bb, solution depth dd, and unit step costs. How many iterations will iterative lengthening require?  
3. Now consider step costs drawn from the continuous range [ϵ,1][ϵ,1], where 0<ϵ<10<ϵ<1. How many iterations are required in the worst case?  
4. Implement the algorithm and apply it to instances of the 8-puzzle and traveling salesperson problems. Compare the algorithm’s performance to that of uniform-cost search, and comment on your results.

[Exercise 22](https://aimacode.github.io/aima-exercises/search-exercises/ex_22/)

Describe a state space in which iterative deepening search performs much worse than depth-first search (for example, O(n2)O(n2) vs. O(n)O(n)).

[Exercise 23](https://aimacode.github.io/aima-exercises/search-exercises/ex_23/)

Write a program that will take as input two Web page URLs and find a path of links from one to the other. What is an appropriate search strategy? Is bidirectional search a good idea? Could a search engine be used to implement a predecessor function?

[Exercise 24 (vacuum-search-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_24/)

Consider the vacuum-world problem defined in .  
1. Which of the algorithms defined in this chapter would be appropriate for this problem? Should the algorithm use tree search or graph search?  
2. Apply your chosen algorithm to compute an optimal sequence of actions for a 3×33×3 world whose initial state has dirt in the three top squares and the agent in the center.  
3. Construct a search agent for the vacuum world, and evaluate its performance in a set of 3×33×3 worlds with probability 0.2 of dirt in each square. Include the search cost as well as path cost in the performance measure, using a reasonable exchange rate.  
4. Compare your best search agent with a simple randomized reflex agent that sucks if there is dirt and otherwise moves randomly.  
5. Consider what would happen if the world were enlarged to n×nn×n. How does the performance of the search agent and of the reflex agent vary with nn?

[Exercise 25 (search-special-case-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_25/)

Prove each of the following statements, or give a counterexample:  
1. Breadth-first search is a special case of uniform-cost search.  
2. Depth-first search is a special case of best-first tree search.  
3. Uniform-cost search is a special case of A search.

[Exercise 26](https://aimacode.github.io/aima-exercises/search-exercises/ex_26/)

Compare the performance of A and RBFS on a set of randomly generated problems in the 8-puzzle (with Manhattan distance) and TSP (with MST—see ) domains. Discuss your results. What happens to the performance of RBFS when a small random number is added to the heuristic values in the 8-puzzle domain?

[Exercise 27](https://aimacode.github.io/aima-exercises/search-exercises/ex_27/)

Trace the operation of A search applied to the problem of getting to Bucharest from Lugoj using the straight-line distance heuristic. That is, show the sequence of nodes that the algorithm will consider and the ff, gg, and hh score for each node.

[Exercise 28](https://aimacode.github.io/aima-exercises/search-exercises/ex_28/)

Sometimes there is no good evaluation function for a problem but there is a good comparison method: a way to tell whether one node is better than another without assigning numerical values to either. Show that this is enough to do a best-first search. Is there an analog of A for this setting?

[Exercise 29 (failure-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_29/)

Devise a state space in which A using returns a suboptimal solution with an h(n)h(n) function that is admissible but inconsistent.

[Exercise 30](https://aimacode.github.io/aima-exercises/search-exercises/ex_30/)

Accurate heuristics don’t necessarily reduce search time in the worst case. Given any depth dd, define a search problem with a goal node at depth dd, and write a heuristic function such that |h(n)−h\\*(n)|≤O(logh\\*(n))|h(n)−h\\*(n)|≤O(log⁡h\\*(n)) but A∗A∗ expands all nodes of depth less than dd.

[Exercise 31](https://aimacode.github.io/aima-exercises/search-exercises/ex_31/)

The **heuristic path algorithm** [Pohl:1977](https://aimacode.github.io/aima-exercises/search-exercises/) is a best-first search in which the evaluation function is f(n)=(2−w)g(n)+wh(n)f(n)=(2−w)g(n)+wh(n). For what values of ww is this complete? For what values is it optimal, assuming that hh is admissible? What kind of search does this perform for w=0w=0, w=1w=1, and w=2w=2?

[Exercise 32](https://aimacode.github.io/aima-exercises/search-exercises/ex_1/)

Consider the unbounded version of the regular 2D grid shown in . The start state is at the origin, (0,0), and the goal state is at (x,y)(x,y).  
1. What is the branching factor bb in this state space?  
2. How many distinct states are there at depth kk (for k>0k>0)?  
3. What is the maximum number of nodes expanded by breadth-first tree search?  
4. What is the maximum number of nodes expanded by breadth-first graph search?  
5. Is h=|u−x|+|v−y|h=|u−x|+|v−y| an admissible heuristic for a state at (u,v)(u,v)? Explain.  
6. How many nodes are expanded by A graph search using hh?  
7. Does hh remain admissible if some links are removed?  
8. Does hh remain admissible if some links are added between nonadjacent states?

[Exercise 33](https://aimacode.github.io/aima-exercises/search-exercises/ex_33/)

nn vehicles occupy squares (1,1)(1,1) through (n,1)(n,1) (i.e., the bottom row) of an n×nn×n grid. The vehicles must be moved to the top row but in reverse order; so the vehicle ii that starts in (i,1)(i,1) must end up in (n−i+1,n)(n−i+1,n). On each time step, every one of the nn vehicles can move one square up, down, left, or right, or stay put; but if a vehicle stays put, one other adjacent vehicle (but not more than one) can hop over it. Two vehicles cannot occupy the same square.  
1. Calculate the size of the state space as a function of nn.  
2. Calculate the branching factor as a function of nn.  
3. Suppose that vehicle ii is at (xi,yi)(xi,yi); write a nontrivial admissible heuristic hihi for the number of moves it will require to get to its goal location (n−i+1,n)(n−i+1,n), assuming no other vehicles are on the grid.  
4. Which of the following heuristics are admissible for the problem of moving all nn vehicles to their destinations? Explain.  
1. ∑ni=1hi∑i=1nhi.  
2. max{h1,…,hn}max{h1,…,hn}.  
3. min{h1,…,hn}min{h1,…,hn}.

[Exercise 34](https://aimacode.github.io/aima-exercises/search-exercises/ex_34/)

Consider the problem of moving kk knights from kk starting squares s1,…,sks1,…,sk to kk goal squares g1,…,gkg1,…,gk, on an unbounded chessboard, subject to the rule that no two knights can land on the same square at the same time. Each action consists of moving *up to* kk knights simultaneously. We would like to complete the maneuver in the smallest number of actions.  
1. What is the maximum branching factor in this state space, expressed as a function of kk?  
2. Suppose hihi is an admissible heuristic for the problem of moving knight ii to goal gigi by itself. Which of the following heuristics are admissible for the kk-knight problem? Of those, which is the best?  
1. min{h1,…,hk}min{h1,…,hk}.  
2. max{h1,…,hk}max{h1,…,hk}.  
3. ∑ki=1hi∑i=1khi.  
3. Repeat (b) for the case where you are allowed to move only one knight at a time.

[Exercise 35](https://aimacode.github.io/aima-exercises/search-exercises/ex_35/)

We saw on page that the straight-line distance heuristic leads greedy best-first search astray on the problem of going from Iasi to Fagaras. However, the heuristic is perfect on the opposite problem: going from Fagaras to Iasi. Are there problems for which the heuristic is misleading in both directions?

[Exercise 36](https://aimacode.github.io/aima-exercises/search-exercises/ex_36/)

Invent a heuristic function for the 8-puzzle that sometimes overestimates, and show how it can lead to a suboptimal solution on a particular problem. (You can use a computer to help if you want.) Prove that if hh never overestimates by more than cc, A using hh returns a solution whose cost exceeds that of the optimal solution by no more than cc.

[Exercise 37 (consistent-heuristic-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_37/)

Prove that if a heuristic is consistent, it must be admissible. Construct an admissible heuristic that is not consistent.

[Exercise 38 (tsp-mst-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_38/)

The traveling salesperson problem (TSP) can be solved with the minimum-spanning-tree (MST) heuristic, which estimates the cost of completing a tour, given that a partial tour has already been constructed. The MST cost of a set of cities is the smallest sum of the link costs of any tree that connects all the cities.  
1. Show how this heuristic can be derived from a relaxed version of the TSP.  
2. Show that the MST heuristic dominates straight-line distance.  
3. Write a problem generator for instances of the TSP where cities are represented by random points in the unit square.  
4. Find an efficient algorithm in the literature for constructing the MST, and use it with A graph search to solve instances of the TSP.

[Exercise 39 (Gaschnig-h-exercise)](https://aimacode.github.io/aima-exercises/search-exercises/ex_39/)

On page , we defined the relaxation of the 8-puzzle in which a tile can move from square A to square B if B is blank. The exact solution of this problem defines **Gaschnig's heuristic** [Gaschnig:1979](https://aimacode.github.io/aima-exercises/search-exercises/). Explain why Gaschnig’s heuristic is at least as accurate as h1h1 (misplaced tiles), and show cases where it is more accurate than both h1h1 and h2h2 (Manhattan distance). Explain how to calculate Gaschnig’s heuristic efficiently.

[Exercise 40](https://aimacode.github.io/aima-exercises/search-exercises/ex_40/)

We gave two simple heuristics for the 8-puzzle: Manhattan distance and misplaced tiles. Several heuristics in the literature purport to improve on this—see, for example, [Nilsson:1971](https://aimacode.github.io/aima-exercises/search-exercises/), [Mostow+Prieditis:1989](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.3333&rep=rep1&type=pdf), and [Hansson+al:1992](https://europepmc.org/abstract/med/1534722). Test these claims by implementing the heuristics and comparing the performance of the resulting algorithms.

**4. Beyond Classical Search**

[Exercise 1](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_1/)

Give the name of the algorithm that results from each of the following special cases:  
1. Local beam search with k=1k=1.  
2. Local beam search with one initial state and no limit on the number of states retained.  
3. Simulated annealing with T=0T=0 at all times (and omitting the termination test).  
4. Simulated annealing with T=∞T=∞ at all times.  
5. Genetic algorithm with population size N=1N=1.

[Exercise 2](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_2/)

Exercise [3.19](https://aimacode.github.io/aima-exercises/search-exercises/ex_19) 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.

[Exercise 3](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_3/)

In this exercise, we explore the use of local search methods to solve TSPs of the type defined in Exercise [3.38](https://aimacode.github.io/aima-exercises/search-exercises/ex_38)  
1. 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 [3.38](https://aimacode.github.io/aima-exercises/search-exercises/ex_38))  
2. Repeat part (a) using a genetic algorithm instead of hill climbing. You may want to consult @Larranaga+al:1999 for some suggestions for representations.

[Exercise 4 (hill-climbing-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_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.

[Exercise 5 (cond-plan-repeated-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_5/)

The **And-Or-Graph-Search** algorithm in Figure [4.11](https://aimacode.github.io/aima-exercises/figures/and-or-graph-search-algorithm.png) 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 in Figure [3.11](https://aimacode.github.io/aima-exercises/advanced-search-exercises/) 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 , to avoid having multiple copies of subplans.

[Exercise 6 (cond-loop-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_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.

[Exercise 7](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_7/)

In Section  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 bb to a goal state. Suppose the agent knows h\\*(s)h\\*(s), the true optimal cost of solving the physical state ss in the fully observable problem, for every state ss in bb. Find an admissible heuristic h(b)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](https://aimacode.github.io/aima-exercises/figures/vacuum2-sets-figure.png). How well does A\* perform?

[Exercise 8 (belief-state-superset-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_8/)

This exercise explores subset–superset relations between belief states in sensorless or partially observable environments.  
1. Prove that if an action sequence is a solution for a belief state bb, it is also a solution for any subset of bb. Can anything be said about supersets of bb?  
2. Explain in detail how to modify graph search for sensorless problems to take advantage of your answers in (a).  
3. Explain in detail how to modify and–or search for partially observable problems, beyond the modifications you describe in (b).

[Exercise 9 (multivalued-sensorless-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_9/)

On page  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.

[Exercise 10 (vacuum-solvable-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_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}{1,2,3,4,5,6,7,8}, and explain why the problem is unsolvable.

[Exercise 11 (vacuum-solvable-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_11/)

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

[Exercise 12 (path-planning-agent-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_12/)

We can turn the navigation problem in Exercise [3.9](https://aimacode.github.io/aima-exercises/search-exercises/ex_9) 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.  
1. 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.  
2. Document your agent’s performance (by having the agent generate suitable commentary as it moves around) and report its performance over 100 episodes.  
3. 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.  
4. 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?  
5. 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?

[Exercise 13 (online-offline-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_13/)

Suppose that an agent is in a 3×33×3 maze environment like the one shown in Figure [4.19](https://aimacode.github.io/aima-exercises/figures/maze-3x3-figure.png). 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.  
1. 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?  
2. How many distinct percepts are possible in the initial state?  
3. 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.

[Exercise 14 (online-offline-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_14/)

Suppose that an agent is in a 3×33×3 maze environment like the one shown in Figure [4.19](https://aimacode.github.io/aima-exercises/figures/maze-3x3-figure.png). The agent knows that its initial location is (3,3), that the goal is at (1,1), and that the four 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 or is a new state.  
1. 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?  
2. How many distinct percepts are possible in the initial state?  
3. 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.

[Exercise 15 (path-planning-hc-exercise)](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_15/)

In this exercise, we examine hill climbing in the context of robot navigation, using the environment in Figure [3.31](https://aimacode.github.io/aima-exercises/figures/geometric-scene-figure.png) as an example.  
1. Repeat Exercise [4.12](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_12) using hill climbing. Does your agent ever get stuck in a local minimum? Is it \*possible\* for it to get stuck with convex obstacles?  
2. Construct a nonconvex polygonal environment in which the agent gets stuck.  
3. Modify the hill-climbing algorithm so that, instead of doing a depth-1 search to decide where to go next, it does a depth-kk search. It should find the best kk-step path and do one step along it, and then repeat the process.  
4. Is there some kk for which the new algorithm is guaranteed to escape from local minima?  
5. Explain how LRTA enables the agent to escape from local minima in this case.

[Exercise 16](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_16/)

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)(1,0), (0,1)(0,1), (−1,0)(−1,0), (0,−1)(0,−1), tried in that order. Show that online DFS starting at (0,0)(0,0) will not reach (1,−1)(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)(1,−1)?

[Exercise 17](https://aimacode.github.io/aima-exercises/advanced-search-exercises/ex_17/)

Relate the time complexity of LRTA\* to its space complexity.

**5. Adversarial Search**

[Exercise 1](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_1/)

Suppose you have an oracle, OM(s)OM(s), that correctly predicts the opponent’s move in any state. Using this, formulate the definition of a game as a (single-agent) search problem. Describe an algorithm for finding the optimal move.

[Exercise 2](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_2/)

Consider the problem of solving two 8-puzzles.  
1. Give a complete problem formulation in the style of Chapter [search-chapter.](https://aimacode.github.io/aima-exercises/search-exercises/)  
2. How large is the reachable state space? Give an exact numerical expression.  
3. Suppose we make the problem adversarial as follows: the two players take turns moving; a coin is flipped to determine the puzzle on which to make a move in that turn; and the winner is the first to solve one puzzle. Which algorithm can be used to choose a move in this setting?  
4. Does the game eventually end, given optimal play? Explain.  
(a) A map where the cost of every edge is 1. Initially the pursuer PP is at node **b** and the evader EE is at node **d**  
(b) A partial game tree for this map. Each node is labeled with the P,EP,E positions. PP moves first. Branches marked "?" have yet to be explored.

**Pursuit evasion game Figure**

[Exercise 3](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_3/)

Imagine that, in Exercise [3.5](https://aimacode.github.io/aima-exercises/search-exercises/ex_5), one of the friends wants to avoid the other. The problem then becomes a two-player game. We assume now that the players take turns moving. The game ends only when the players are on the same node; the terminal payoff to the pursuer is minus the total time taken. (The evader “wins” by never losing.) An example is shown in Figure. [pursuit-evasion-game-figure](https://aimacode.github.io/aima-exercises/game-playing-exercises/#pursuit-evasion-game-figure)  
1. Copy the game tree and mark the values of the terminal nodes.  
2. Next to each internal node, write the strongest fact you can infer about its value (a number, one or more inequalities such as “≥14≥14”, or a “?”).  
3. Beneath each question mark, write the name of the node reached by that branch.  
4. Explain how a bound on the value of the nodes in (c) can be derived from consideration of shortest-path lengths on the map, and derive such bounds for these nodes. Remember the cost to get to each leaf as well as the cost to solve it.  
5. Now suppose that the tree as given, with the leaf bounds from (d), is evaluated from left to right. Circle those “?” nodes that would *not* need to be expanded further, given the bounds from part (d), and cross out those that need not be considered at all.  
6. Can you prove anything in general about who wins the game on a map that is a tree?

[Exercise 4 (game-playing-chance-exercise)](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_4/)

Describe and implement state descriptions, move generators, terminal tests, utility functions, and evaluation functions for one or more of the following stochastic games: Monopoly, Scrabble, bridge play with a given contract, or Texas hold’em poker.

[Exercise 5](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_5/)

Describe and implement a *real-time*, *multiplayer* game-playing environment, where time is part of the environment state and players are given fixed time allocations.

[Exercise 6](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_6/)

Discuss how well the standard approach to game playing would apply to games such as tennis, pool, and croquet, which take place in a continuous physical state space.

[Exercise 7 (minimax-optimality-exercise)](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_7/)

Prove the following assertion: For every game tree, the utility obtained by max using minimax decisions against a suboptimal min will never be lower than the utility obtained playing against an optimal min. Can you come up with a game tree in which max can do still better using a *suboptimal* strategy against a suboptimal min?  
Player AA moves first. The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if AA is on 3 and BB is on 2, then AA may move back to 1.) The game ends when one player reaches the opposite end of the board. If player AA reaches space 4 first, then the value of the game to AA is +1+1; if player BB reaches space 1 first, then the value of the game to AA is −1−1.

**The starting position of a simple game.**

[Exercise 8](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_8/)

Consider the two-player game described in Figure   
1. Draw the complete game tree, using the following conventions:  
- Write each state as (sA,sB)(sA,sB), where sAsA and sBsB denote the token locations.  
- Put each terminal state in a square box and write its game value in a circle.  
- Put *loop states* (states that already appear on the path to the root) in double square boxes. Since their value is unclear, annotate each with a “?” in a circle.  
2. Now mark each node with its backed-up minimax value (also in a circle). Explain how you handled the “?” values and why.  
3. Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to (b). Does your modified algorithm give optimal decisions for all games with loops?  
4. This 4-square game can be generalized to nn squares for any n>2n>2. Prove that AA wins if nn is even and loses if nn is odd.

[Exercise 9](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_9/)

This problem exercises the basic concepts of game playing, using tic-tac-toe (noughts and crosses) as an example. We define XnXn as the number of rows, columns, or diagonals with exactly nn XX’s and no OO’s. Similarly, OnOn is the number of rows, columns, or diagonals with just nn OO’s. The utility function assigns +1+1 to any position with X3=1X3=1 and −1−1 to any position with O3=1O3=1. All other terminal positions have utility 0. For nonterminal positions, we use a linear evaluation function defined as Eval(s)=3X2(s)+X1(s)−(3O2(s)+O1(s))Eval(s)=3X2(s)+X1(s)−(3O2(s)+O1(s)).  
1. Approximately how many possible games of tic-tac-toe are there?  
2. Show the whole game tree starting from an empty board down to depth 2 (i.e., one XX and one OO on the board), taking symmetry into account.  
3. Mark on your tree the evaluations of all the positions at depth 2.  
4. Using the minimax algorithm, mark on your tree the backed-up values for the positions at depths 1 and 0, and use those values to choose the best starting move.  
5. Circle the nodes at depth 2 that would *not* be evaluated if alpha–beta pruning were applied, assuming the nodes are generated in the optimal order for alpha–beta pruning.

[Exercise 10](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_10/)

Consider the family of generalized tic-tac-toe games, defined as follows. Each particular game is specified by a set SS of *squares* and a collection WW of *winning positions.* Each winning position is a subset of SS. For example, in standard tic-tac-toe, SS is a set of 9 squares and WW is a collection of 8 subsets of WW: the three rows, the three columns, and the two diagonals. In other respects, the game is identical to standard tic-tac-toe. Starting from an empty board, players alternate placing their marks on an empty square. A player who marks every square in a winning position wins the game. It is a tie if all squares are marked and neither player has won.  
1. Let N=|S|N=|S|, the number of squares. Give an upper bound on the number of nodes in the complete game tree for generalized tic-tac-toe as a function of NN.  
2. Give a lower bound on the size of the game tree for the worst case, where W={}W={}.  
3. Propose a plausible evaluation function that can be used for any instance of generalized tic-tac-toe. The function may depend on SS and WW.  
4. Assume that it is possible to generate a new board and check whether it is a winning position in 100NN machine instructions and assume a 2 gigahertz processor. Ignore memory limitations. Using your estimate in (a), roughly how large a game tree can be completely solved by alpha–beta in a second of CPU time? a minute? an hour?

[Exercise 11](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_11/)

Develop a general game-playing program, capable of playing a variety of games.  
1. Implement move generators and evaluation functions for one or more of the following games: Kalah, Othello, checkers, and chess.  
2. Construct a general alpha–beta game-playing agent.  
3. Compare the effect of increasing search depth, improving move ordering, and improving the evaluation function. How close does your effective branching factor come to the ideal case of perfect move ordering?  
4. Implement a selective search algorithm, such as B\\* [Berliner:1979](https://aimacode.github.io/aima-exercises/game-playing-exercises/), conspiracy number search @McAllester:1988, or MGSS\\* [Russell+Wefald:1989](https://aimacode.github.io/aima-exercises/game-playing-exercises/) and compare its performance to A\\*.

[Exercise 12](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_12/)

Describe how the minimax and alpha–beta algorithms change for two-player, non-zero-sum games in which each player has a distinct utility function and both utility functions are known to both players. If there are no constraints on the two terminal utilities, is it possible for any node to be pruned by alpha–beta? What if the player’s utility functions on any state differ by at most a constant kk, making the game almost cooperative?

[Exercise 13](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_13/)

Describe how the minimax and alpha–beta algorithms change for two-player, non-zero-sum games in which each player has a distinct utility function and both utility functions are known to both players. If there are no constraints on the two terminal utilities, is it possible for any node to be pruned by alpha–beta? What if the player’s utility functions on any state sum to a number between constants −k−k and kk, making the game almost zero-sum?

[Exercise 14](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_14/)

Develop a formal proof of correctness for alpha–beta pruning. To do this, consider the situation shown in Figure . The question is whether to prune node njnj, which is a max-node and a descendant of node n1n1. The basic idea is to prune it if and only if the minimax value of n1n1 can be shown to be independent of the value of njnj.  
1. Mode n1n1 takes on the minimum value among its children: n1=min(n2,n21,…,n2b2)n1=min(n2,n21,…,n2b2). Find a similar expression for n2n2 and hence an expression for n1n1 in terms of njnj.  
2. Let lili be the minimum (or maximum) value of the nodes to the *left* of node nini at depth ii, whose minimax value is already known. Similarly, let riri be the minimum (or maximum) value of the unexplored nodes to the right of nini at depth ii. Rewrite your expression for n1n1 in terms of the lili and riri values.  
3. Now reformulate the expression to show that in order to affect n1n1, njnj must not exceed a certain bound derived from the lili values.  
4. Repeat the process for the case where njnj is a min-node.

**Situation when considering whether to prune node**njnj**.**

[Exercise 15](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_15/)

Prove that the alpha–beta algorithm takes time O(bm/2)O(bm/2) with optimal move ordering, where mm is the maximum depth of the game tree.

[Exercise 16](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_16/)

Suppose you have a chess program that can evaluate 5 million nodes per second. Decide on a compact representation of a game state for storage in a transposition table. About how many entries can you fit in a 1-gigabyte in-memory table? Will that be enough for the three minutes of search allocated for one move? How many table lookups can you do in the time it would take to do one evaluation? Now suppose the transposition table is stored on disk. About how many evaluations could you do in the time it takes to do one disk seek with standard disk hardware?

[Exercise 17](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_17/)

Suppose you have a chess program that can evaluate 10 million nodes per second. Decide on a compact representation of a game state for storage in a transposition table. About how many entries can you fit in a 2-gigabyte in-memory table? Will that be enough for the three minutes of search allocated for one move? How many table lookups can you do in the time it would take to do one evaluation? Now suppose the transposition table is stored on disk. About how many evaluations could you do in the time it takes to do one disk seek with standard disk hardware?

**The complete game tree for a trivial game with chance nodes..**

[Exercise 18](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_18/)

This question considers pruning in games with chance nodes. Figure  shows the complete game tree for a trivial game. Assume that the leaf nodes are to be evaluated in left-to-right order, and that before a leaf node is evaluated, we know nothing about its value—the range of possible values is −∞−∞ to ∞∞.  
1. Copy the figure, mark the value of all the internal nodes, and indicate the best move at the root with an arrow.  
2. Given the values of the first six leaves, do we need to evaluate the seventh and eighth leaves? Given the values of the first seven leaves, do we need to evaluate the eighth leaf? Explain your answers.  
3. Suppose the leaf node values are known to lie between –2 and 2 inclusive. After the first two leaves are evaluated, what is the value range for the left-hand chance node?  
4. Circle all the leaves that need not be evaluated under the assumption in (c).

[Exercise 19](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_19/)

Implement the expectiminimax algorithm and the \\*-alpha–beta algorithm, which is described by [Ballard:1983](https://aimacode.github.io/aima-exercises/game-playing-exercises/), for pruning game trees with chance nodes. Try them on a game such as backgammon and measure the pruning effectiveness of \\*-alpha–beta.

[Exercise 20 (game-linear-transform)](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_20/)

Prove that with a positive linear transformation of leaf values (i.e., transforming a value xx to ax+bax+b where a>0a>0), the choice of move remains unchanged in a game tree, even when there are chance nodes.

[Exercise 21 (game-playing-monte-carlo-exercise)](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_21/)

Consider the following procedure for choosing moves in games with chance nodes:  
- Generate some dice-roll sequences (say, 50) down to a suitable depth (say, 8).  
- With known dice rolls, the game tree becomes deterministic. For each dice-roll sequence, solve the resulting deterministic game tree using alpha–beta.  
- Use the results to estimate the value of each move and to choose the best.  
Will this procedure work well? Why (or why not)?

[Exercise 22](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_22/)

In the following, a “max” tree consists only of max nodes, whereas an “expectimax” tree consists of a max node at the root with alternating layers of chance and max nodes. At chance nodes, all outcome probabilities are nonzero. The goal is to *find the value of the root* with a bounded-depth search. For each of (a)–(f), either give an example or explain why this is impossible.  
1. Assuming that leaf values are finite but unbounded, is pruning (as in alpha–beta) ever possible in a max tree?  
2. Is pruning ever possible in an expectimax tree under the same conditions?  
3. If leaf values are all nonnegative, is pruning ever possible in a max tree? Give an example, or explain why not.  
4. If leaf values are all nonnegative, is pruning ever possible in an expectimax tree? Give an example, or explain why not.  
5. If leaf values are all in the range [0,1][0,1], is pruning ever possible in a max tree? Give an example, or explain why not.  
6. If leaf values are all in the range [0,1][0,1], is pruning ever possible in an expectimax tree?1  
7. Consider the outcomes of a chance node in an expectimax tree. Which of the following evaluation orders is most likely to yield pruning opportunities?  
i. Lowest probability first  
ii. Highest probability first  
iii. Doesn’t make any difference

[Exercise 23](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_23/)

In the following, a “max” tree consists only of max nodes, whereas an “expectimax” tree consists of a max node at the root with alternating layers of chance and max nodes. At chance nodes, all outcome probabilities are nonzero. The goal is to *find the value of the root* with a bounded-depth search.  
1. Assuming that leaf values are finite but unbounded, is pruning (as in alpha–beta) ever possible in a max tree? Give an example, or explain why not.  
2. Is pruning ever possible in an expectimax tree under the same conditions? Give an example, or explain why not.  
3. If leaf values are constrained to be in the range [0,1][0,1], is pruning ever possible in a max tree? Give an example, or explain why not.  
4. If leaf values are constrained to be in the range [0,1][0,1], is pruning ever possible in an expectimax tree? Give an example (qualitatively different from your example in (e), if any), or explain why not.  
5. If leaf values are constrained to be nonnegative, is pruning ever possible in a max tree? Give an example, or explain why not.  
6. If leaf values are constrained to be nonnegative, is pruning ever possible in an expectimax tree? Give an example, or explain why not.  
7. Consider the outcomes of a chance node in an expectimax tree. Which of the following evaluation orders is most likely to yield pruning opportunities: (i) Lowest probability first; (ii) Highest probability first; (iii) Doesn’t make any difference?

[Exercise 24](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_24/)

Suppose you have an oracle, OM(s)OM(s), that correctly predicts the opponent’s move in any state. Using this, formulate the definition of a game as a (single-agent) search problem. Describe an algorithm for finding the optimal move.

[Exercise 25](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_25/)

Consider carefully the interplay of chance events and partial information in each of the games in Exercise [5.4](https://aimacode.github.io/aima-exercises/game-playing-exercises/ex_4).  
1. For which is the standard expectiminimax model appropriate? Implement the algorithm and run it in your game-playing agent, with appropriate modifications to the game-playing environment.  
2. For which would the scheme described in Exercise [game-playing-monte-carlo-exercise](https://aimacode.github.io/aima-exercises/game-playing-exercises/#ex5.21) be appropriate?  
3. Discuss how you might deal with the fact that in some of the games, the players do not have the same knowledge of the current state.

**6. Constraint Satisfaction Problems**

[Exercise 1](https://aimacode.github.io/aima-exercises/csp-exercises/ex_1/)

How many solutions are there for the map-coloring problem in Figure [6.1](https://aimacode.github.io/aima-exercises/figures/australia-figure.png)? How many solutions if four colors are allowed? Two colors?

[Exercise 2](https://aimacode.github.io/aima-exercises/csp-exercises/ex_2/)

Consider the problem of placing kk knights on an n×nn×n chessboard such that no two knights are attacking each other, where kk is given and k≤n2k≤n2.  
1. Choose a CSP formulation. In your formulation, what are the variables?  
2. What are the possible values of each variable?  
3. What sets of variables are constrained, and how?  
4. Now consider the problem of putting \*as many knights as possible\* on the board without any attacks. Explain how to solve this with local search by defining appropriate ACTIONS and RESULT functions and a sensible objective function.

[Exercise 3 (crossword-exercise)](https://aimacode.github.io/aima-exercises/csp-exercises/ex_3/)

Consider the problem of [constructing](https://aimacode.github.io/aima-exercises/csp-exercises/#footnote1) (not solving) crossword puzzles fitting words into a rectangular grid. The grid, which is given as part of the problem, specifies which squares are blank and which are shaded. Assume that a list of words (i.e., a dictionary) is provided and that the task is to fill in the blank squares by using any subset of the list. Formulate this problem precisely in two ways:  
1. As a general search problem. Choose an appropriate search algorithm and specify a heuristic function. Is it better to fill in blanks one letter at a time or one word at a time?  
2. As a constraint satisfaction problem. Should the variables be words or letters?  
Which formulation do you think will be better? Why?

[Exercise 4 (csp-definition-exercise)](https://aimacode.github.io/aima-exercises/csp-exercises/ex_4/)

Give precise formulations for each of the following as constraint satisfaction problems:  
1. Rectilinear floor-planning: find non-overlapping places in a large rectangle for a number of smaller rectangles.  
2. Class scheduling: There is a fixed number of professors and classrooms, a list of classes to be offered, and a list of possible time slots for classes. Each professor has a set of classes that he or she can teach.  
3. Hamiltonian tour: given a network of cities connected by roads, choose an order to visit all cities in a country without repeating any.

[Exercise 5](https://aimacode.github.io/aima-exercises/csp-exercises/ex_5/)

Solve the cryptarithmetic problem in Figure [6.2](https://aimacode.github.io/aima-exercises/figures/cryptarithmetic-figure.png) by hand, using the strategy of backtracking with forward checking and the MRV and least-constraining-value heuristics.

[Exercise 6 (nary-csp-exercise)](https://aimacode.github.io/aima-exercises/csp-exercises/ex_6/)

Show how a single ternary constraint such as “A+B=CA+B=C” can be turned into three binary constraints by using an auxiliary variable. You may assume finite domains. (\*Hint:\* Consider a new variable that takes on values that are pairs of other values, and consider constraints such as “XX is the first element of the pair YY.”) Next, show how constraints with more than three variables can be treated similarly. Finally, show how unary constraints can be eliminated by altering the domains of variables. This completes the demonstration that any CSP can be transformed into a CSP with only binary constraints.

[Exercise 7 (zebra-exercise)](https://aimacode.github.io/aima-exercises/csp-exercises/ex_7/)

Consider the following logic puzzle: In five houses, each with a different color, live five persons of different nationalities, each of whom prefers a different brand of candy, a different drink, and a different pet. Given the following facts, the questions to answer are “Where does the zebra live, and in which house do they drink water?”  
The Englishman lives in the red house.  
The Spaniard owns the dog.  
The Norwegian lives in the first house on the left.  
The green house is immediately to the right of the ivory house.  
The man who eats Hershey bars lives in the house next to the man with the fox.  
Kit Kats are eaten in the yellow house.  
The Norwegian lives next to the blue house.  
The Smarties eater owns snails.  
The Snickers eater drinks orange juice.  
The Ukrainian drinks tea.  
The Japanese eats Milky Ways.  
Kit Kats are eaten in a house next to the house where the horse is kept.  
Coffee is drunk in the green house.  
Milk is drunk in the middle house.  
Discuss different representations of this problem as a CSP. Why would one prefer one representation over another?

[Exercise 8](https://aimacode.github.io/aima-exercises/csp-exercises/ex_8/)

Consider the graph with 8 nodes A1A1, A2A2, A3A3, A4A4, HH, TT, F1F1, F2F2. AiAi is connected to Ai+1Ai+1 for all ii, each AiAi is connected to HH, HH is connected to TT, and TT is connected to each FiFi. Find a 3-coloring of this graph by hand using the following strategy: backtracking with conflict-directed backjumping, the variable order A1A1, HH, A4A4, F1F1, A2A2, F2F2, A3A3, TT, and the value order RR, GG, BB.

[Exercise 9](https://aimacode.github.io/aima-exercises/csp-exercises/ex_9/)

Explain why it is a good heuristic to choose the variable that is \*most\* constrained but the value that is \*least\* constraining in a CSP search.

[Exercise 10](https://aimacode.github.io/aima-exercises/csp-exercises/ex_10/)

Generate random instances of map-coloring problems as follows: scatter nn points on the unit square; select a point XX at random, connect XX by a straight line to the nearest point YY such that XX is not already connected to YY and the line crosses no other line; repeat the previous step until no more connections are possible. The points represent regions on the map and the lines connect neighbors. Now try to find kk-colorings of each map, for both k3k3 and k4k4, using min-conflicts, backtracking, backtracking with forward checking, and backtracking with MAC. Construct a table of average run times for each algorithm for values of nn up to the largest you can manage. Comment on your results.

[Exercise 11](https://aimacode.github.io/aima-exercises/csp-exercises/ex_11/)

Use the AC-3 algorithm to show that arc consistency can detect the inconsistency of the partial assignment green,Vredgreen,Vred for the problem shown in Figure [6.1](https://aimacode.github.io/aima-exercises/figures/australia-figure.png).

[Exercise 12](https://aimacode.github.io/aima-exercises/csp-exercises/ex_12/)

Use the AC-3 algorithm to show that arc consistency can detect the inconsistency of the partial assignment red,Vbluered,Vblue for the problem shown in Figure [6.1](https://aimacode.github.io/aima-exercises/figures/australia-figure.png).

[Exercise 13](https://aimacode.github.io/aima-exercises/csp-exercises/ex_13/)

What is the worst-case complexity of running AC-3 on a tree-structured CSP?

[Exercise 14 (ac4-exercise)](https://aimacode.github.io/aima-exercises/csp-exercises/ex14/)

AC-3 puts back on the queue *every* arc (Xk,XiXk,Xi) whenever *any* value is deleted from the domain of XiXi, even if each value of XkXk is consistent with several remaining values of XiXi. Suppose that, for every arc (Xk,XiXk,Xi), we keep track of the number of remaining values of XiXi that are consistent with each value of XkXk. Explain how to update these numbers efficiently and hence show that arc consistency can be enforced in total time O(n2d2)O(n2d2).

[Exercise 15](https://aimacode.github.io/aima-exercises/csp-exercises/ex15/)

The Tree-CSP-Solver (Figure [6.10](https://aimacode.github.io/aima-exercises/figures/tree-csp-figure.png)) makes arcs consistent starting at the leaves and working backwards towards the root. Why does it do that? What would happen if it went in the opposite direction?

[Exercise 16](https://aimacode.github.io/aima-exercises/csp-exercises/ex16/)

We introduced Sudoku as a CSP to be solved by search over partial assignments because that is the way people generally undertake solving Sudoku problems. It is also possible, of course, to attack these problems with local search over complete assignments. How well would a local solver using the min-conflicts heuristic do on Sudoku problems?

[Exercise 17](https://aimacode.github.io/aima-exercises/csp-exercises/ex17/)

Define in your own words the terms constraint, backtracking search, arc consistency, backjumping, min-conflicts, and cycle cutset.

[Exercise 18](https://aimacode.github.io/aima-exercises/csp-exercises/ex18/)

Define in your own words the terms constraint, commutativity, arc consistency, backjumping, min-conflicts, and cycle cutset.

[Exercise 19](https://aimacode.github.io/aima-exercises/csp-exercises/ex19/)

Suppose that a graph is known to have a cycle cutset of no more than kk nodes. Describe a simple algorithm for finding a minimal cycle cutset whose run time is not much more than O(nk)O(nk) for a CSP with nn variables. Search the literature for methods for finding approximately minimal cycle cutsets in time that is polynomial in the size of the cutset. Does the existence of such algorithms make the cycle cutset method practical?

[Exercise 20](https://aimacode.github.io/aima-exercises/csp-exercises/ex20/)

Consider the problem of tiling a surface (completely and exactly covering it) with nn dominoes (2×12×1 rectangles). The surface is an arbitrary edge-connected (i.e., adjacent along an edge, not just a corner) collection of 2n2n 1×11×1 squares (e.g., a checkerboard, a checkerboard with some squares missing, a 10×110×1 row of squares, etc.).  
1. Formulate this problem precisely as a CSP where the dominoes are the variables.  
2. Formulate this problem precisely as a CSP where the squares are the variables, keeping the state space as small as possible. (\*Hint:\* does it matter which particular domino goes on a given pair of squares?)  
3. Construct a surface consisting of 6 squares such that your CSP formulation from part (b) has a \*tree-structured\* constraint graph.  
4. Describe exactly the set of solvable instances that have a tree-structured constraint graph.

^1. @Ginsberg+al:1990 discuss several methods for constructing crossword puzzles. @Littman+al:1999 tackle the harder problem of solving them.

7. Logical Agents

[Exercise 1](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_1/)

Suppose the agent has progressed to the point shown in Figure [7.4](https://aimacode.github.io/aima-exercises/figures/wumpus-seq35-figure.png)(a), page , having perceived nothing in [1,1], a breeze in [2,1], and a stench in [1,2], and is now concerned with the contents of [1,3], [2,2], and [3,1]. Each of these can contain a pit, and at most one can contain a wumpus. Following the example of Figure [7.5](https://aimacode.github.io/aima-exercises/figures/wumpus-entailment-figure.png), construct the set of possible worlds. (You should find 32 of them.) Mark the worlds in which the KB is true and those in which each of the following sentences is true:  
α2α2 = “There is no pit in [2,2].”  
α3α3 = “There is a wumpus in [1,3].”  
Hence show that KB⊨α2KB⊨α2 and KB⊨α3KB⊨α3.

[Exercise 2](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_2/)

(Adapted from [Barwise+Etchemendy:1993](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/) .) Given the following, can you prove that the unicorn is mythical? How about magical? Horned?  
Note: If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

[Exercise 3 (truth-value-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_3/)

Consider the problem of deciding whether a propositional logic sentence is true in a given model.  
1. Write a recursive algorithm PL-True?(s,m)(s,m) that returns truetrue if and only if the sentence ss is true in the model mm (where mm assigns a truth value for every symbol in ss). The algorithm should run in time linear in the size of the sentence. (Alternatively, use a version of this function from the online code repository.)  
2. Give three examples of sentences that can be determined to be true or false in a *partial* model that does not specify a truth value for some of the symbols.  
3. Show that the truth value (if any) of a sentence in a partial model cannot be determined efficiently in general.  
4. Modify your algorithm so that it can sometimes judge truth from partial models, while retaining its recursive structure and linear run time. Give three examples of sentences whose truth in a partial model is *not* detected by your algorithm.  
5. Investigate whether the modified algorithm makes TT−Entails?TT−Entails? more efficient.

[Exercise 4](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_4/)

Which of the following are correct?  
1. False⊨TrueFalse⊨True.  
2. True⊨FalseTrue⊨False.  
3. (A∧B)⊨(A⇔B)(A∧B)⊨(A⇔B).  
4. A⇔B⊨A∨BA⇔B⊨A∨B.  
5. A⇔B⊨¬A∨BA⇔B⊨¬A∨B.  
6. (A∧B)⇒C⊨(A⇒C)∨(B⇒C)(A∧B)⇒C⊨(A⇒C)∨(B⇒C).  
7. (C∨(¬A∧¬B))≡((A⇒C)∧(B⇒C))(C∨(¬A∧¬B))≡((A⇒C)∧(B⇒C)).  
8. (A∨B)∧(¬C∨¬D∨E)⊨(A∨B)(A∨B)∧(¬C∨¬D∨E)⊨(A∨B).  
9. (A∨B)∧(¬C∨¬D∨E)⊨(A∨B)∧(¬D∨E)(A∨B)∧(¬C∨¬D∨E)⊨(A∨B)∧(¬D∨E).  
10. (A∨B)∧¬(A⇒B)(A∨B)∧¬(A⇒B) is satisfiable.  
11. (A⇔B)∧(¬A∨B)(A⇔B)∧(¬A∨B) is satisfiable.  
12. (A⇔B)⇔C(A⇔B)⇔C has the same number of models as (A⇔B)(A⇔B) for any fixed set of proposition symbols that includes AA, BB, CC.

[Exercise 5](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_5/)

Which of the following are correct?  
1. False⊨TrueFalse⊨True.  
2. True⊨FalseTrue⊨False.  
3. (A∧B)⊨(A⇔B)(A∧B)⊨(A⇔B).  
4. A⇔B⊨A∨BA⇔B⊨A∨B.  
5. A⇔B⊨¬A∨BA⇔B⊨¬A∨B.  
6. (A∨B)∧(¬C∨¬D∨E)⊨(A∨B∨C)∧(B∧C∧D⇒E)(A∨B)∧(¬C∨¬D∨E)⊨(A∨B∨C)∧(B∧C∧D⇒E).  
7. (A∨B)∧(¬C∨¬D∨E)⊨(A∨B)∧(¬D∨E)(A∨B)∧(¬C∨¬D∨E)⊨(A∨B)∧(¬D∨E).  
8. (A∨B)∧¬(A⇒B)(A∨B)∧¬(A⇒B) is satisfiable.  
9. (A∧B)⇒C⊨(A⇒C)∨(B⇒C)(A∧B)⇒C⊨(A⇒C)∨(B⇒C).  
10. (C∨(¬A∧¬B))≡((A⇒C)∧(B⇒C))(C∨(¬A∧¬B))≡((A⇒C)∧(B⇒C)).  
11. (A⇔B)∧(¬A∨B)(A⇔B)∧(¬A∨B) is satisfiable.  
12. (A⇔B)⇔C(A⇔B)⇔C has the same number of models as (A⇔B)(A⇔B) for any fixed set of proposition symbols that includes AA, BB, CC.

[Exercise 6 (deduction-theorem-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_6/)

Prove each of the following assertions:  
1. αα is valid if and only if True⊨αTrue⊨α.  
2. For any αα, False⊨αFalse⊨α.  
3. α⊨βα⊨β if and only if the sentence (α⇒β)(α⇒β) is valid.  
4. α≡βα≡β if and only if the sentence (α⇔β)(α⇔β) is valid.  
5. α⊨βα⊨β if and only if the sentence (α∧¬β)(α∧¬β) is unsatisfiable.

[Exercise 7](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_7/)

Prove, or find a counterexample to, each of the following assertions:  
1. If α⊨γα⊨γ or β⊨γβ⊨γ (or both) then (α∧β)⊨γ(α∧β)⊨γ  
2. If (α∧β)⊨γ(α∧β)⊨γ then α⊨γα⊨γ or β⊨γβ⊨γ (or both).  
3. If α⊨(β∨γ)α⊨(β∨γ) then α⊨βα⊨β or α⊨γα⊨γ (or both).

[Exercise 8](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_8/)

Prove, or find a counterexample to, each of the following assertions:  
1. If α⊨γα⊨γ or β⊨γβ⊨γ (or both) then (α∧β)⊨γ(α∧β)⊨γ  
2. If α⊨(β∧γ)α⊨(β∧γ) then α⊨βα⊨β and α⊨γα⊨γ.  
3. If α⊨(β∨γ)α⊨(β∨γ) then α⊨βα⊨β or α⊨γα⊨γ (or both).

[Exercise 9](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_9/)

Consider a vocabulary with only four propositions, AA, BB, CC, and DD. How many models are there for the following sentences?  
1. B∨CB∨C.  
2. ¬A∨¬B∨¬C∨¬D¬A∨¬B∨¬C∨¬D.  
3. (A⇒B)∧A∧¬B∧C∧D(A⇒B)∧A∧¬B∧C∧D.

[Exercise 10](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_10/)

We have defined four binary logical connectives.  
1. Are there any others that might be useful?  
2. How many binary connectives can there be?  
3. Why are some of them not very useful?

[Exercise 11 (logical-equivalence-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_11/)

Using a method of your choice, verify each of the equivalences in Table [7.11](https://aimacode.github.io/aima-exercises/figures/logical-equivalence-table.png) (page ).

[Exercise 12 (propositional-validity-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_12/)

Decide whether each of the following sentences is valid, unsatisfiable, or neither. Verify your decisions using truth tables or the equivalence rules of Table [7.11](https://aimacode.github.io/aima-exercises/figures/logical-equivalence-table.png) (page ). 1. Smoke⇒SmokeSmoke⇒Smoke  
2. Smoke⇒FireSmoke⇒Fire  
3. (Smoke⇒Fire)⇒(¬Smoke⇒¬Fire)(Smoke⇒Fire)⇒(¬Smoke⇒¬Fire)  
4. Smoke∨Fire∨¬FireSmoke∨Fire∨¬Fire  
5. ((Smoke∧Heat)⇒Fire)⇔((Smoke⇒Fire)∨(Heat⇒Fire))((Smoke∧Heat)⇒Fire)⇔((Smoke⇒Fire)∨(Heat⇒Fire))  
6. (Smoke⇒Fire)⇒((Smoke∧Heat)⇒Fire)(Smoke⇒Fire)⇒((Smoke∧Heat)⇒Fire)  
7. Big∨Dumb∨(Big⇒Dumb)Big∨Dumb∨(Big⇒Dumb)

[Exercise 13 (propositional-validity-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_13/)

Decide whether each of the following sentences is valid, unsatisfiable, or neither. Verify your decisions using truth tables or the equivalence rules of Table [7.11](https://aimacode.github.io/aima-exercises/figures/logical-equivalence-table.png) (page ).  
1. Smoke⇒SmokeSmoke⇒Smoke  
2. Smoke⇒FireSmoke⇒Fire  
3. (Smoke⇒Fire)⇒(¬Smoke⇒¬Fire)(Smoke⇒Fire)⇒(¬Smoke⇒¬Fire)  
4. Smoke∨Fire∨¬FireSmoke∨Fire∨¬Fire  
5. ((Smoke∧Heat)⇒Fire)⇔((Smoke⇒Fire)∨(Heat⇒Fire))((Smoke∧Heat)⇒Fire)⇔((Smoke⇒Fire)∨(Heat⇒Fire))  
6. Big∨Dumb∨(Big⇒Dumb)Big∨Dumb∨(Big⇒Dumb)  
7. (Big∧Dumb)∨¬Dumb(Big∧Dumb)∨¬Dumb

[Exercise 14 (cnf-proof-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_14/)

Any propositional logic sentence is logically equivalent to the assertion that each possible world in which it would be false is not the case. From this observation, prove that any sentence can be written in CNF.

[Exercise 15](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_15/)

Use resolution to prove the sentence ¬A∧¬B¬A∧¬B from the clauses in Exercise [7.25](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_25).

[Exercise 16 (inf-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_16/)

This exercise looks into the relationship between clauses and implication sentences.  
1. Show that the clause (¬P1∨⋯∨¬Pm∨Q)(¬P1∨⋯∨¬Pm∨Q) is logically equivalent to the implication sentence (P1∧⋯∧Pm)⇒Q(P1∧⋯∧Pm)⇒Q.  
2. Show that every clause (regardless of the number of positive literals) can be written in the form (P1∧⋯∧Pm)⇒(Q1∨⋯∨Qn)(P1∧⋯∧Pm)⇒(Q1∨⋯∨Qn), where the PPs and QQs are proposition symbols. A knowledge base consisting of such sentences is in implicative normal form or **Kowalski form** [Kowalski:1979](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/).  
3. Write down the full resolution rule for sentences in implicative normal form.

[Exercise 17](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_17/)

According to some political pundits, a person who is radical (RR) is electable (EE) if he/she is conservative (CC), but otherwise is not electable.  
1. Which of the following are correct representations of this assertion?  
1. (R∧E)⟺C(R∧E)⟺C  
2. R⇒(E⟺C)R⇒(E⟺C)  
3. R⇒((C⇒E)∨¬E)R⇒((C⇒E)∨¬E)  
2. Which of the sentences in (a) can be expressed in Horn form?

[Exercise 18](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_18/)

This question considers representing satisfiability (SAT) problems as CSPs.  
1. Draw the constraint graph corresponding to the SAT problem

(¬X1∨X2)∧(¬X2∨X3)∧…∧(¬Xn−1∨Xn)(¬X1∨X2)∧(¬X2∨X3)∧…∧(¬Xn−1∨Xn)

for the particular case n5n5.  
2. How many solutions are there for this general SAT problem as a function of nn?  
3. Suppose we apply {Backtracking-Search} (page ) to find *all* solutions to a SAT CSP of the type given in (a). (To find *all* solutions to a CSP, we simply modify the basic algorithm so it continues searching after each solution is found.) Assume that variables are ordered X1,…,XnX1,…,Xn and falsefalse is ordered before truetrue. How much time will the algorithm take to terminate? (Write an O(⋅)O(⋅) expression as a function of nn.)  
4. We know that SAT problems in Horn form can be solved in linear time by forward chaining (unit propagation). We also know that every tree-structured binary CSP with discrete, finite domains can be solved in time linear in the number of variables (Section ). Are these two facts connected? Discuss.

[Exercise 19](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_19/)

This question considers representing satisfiability (SAT) problems as CSPs.  
1. Draw the constraint graph corresponding to the SAT problem

(¬X1∨X2)∧(¬X2∨X3)∧…∧(¬Xn−1∨Xn)(¬X1∨X2)∧(¬X2∨X3)∧…∧(¬Xn−1∨Xn)

for the particular case n4n4.  
2. How many solutions are there for this general SAT problem as a function of nn?  
3. Suppose we apply {Backtracking-Search} (page ) to find *all* solutions to a SAT CSP of the type given in (a). (To find *all* solutions to a CSP, we simply modify the basic algorithm so it continues searching after each solution is found.) Assume that variables are ordered X1,…,XnX1,…,Xn and falsefalse is ordered before truetrue. How much time will the algorithm take to terminate? (Write an O(⋅)O(⋅) expression as a function of nn.)  
4. We know that SAT problems in Horn form can be solved in linear time by forward chaining (unit propagation). We also know that every tree-structured binary CSP with discrete, finite domains can be solved in time linear in the number of variables (Section ). Are these two facts connected? Discuss.

[Exercise 20](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_20/)

Explain why every nonempty propositional clause, by itself, is satisfiable. Prove rigorously that every set of five 3-SAT clauses is satisfiable, provided that each clause mentions exactly three distinct variables. What is the smallest set of such clauses that is unsatisfiable? Construct such a set.

[Exercise 21](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_21/)

A propositional *2-CNF* expression is a conjunction of clauses, each containing *exactly 2* literals, e.g.,

(A∨B)∧(¬A∨C)∧(¬B∨D)∧(¬C∨G)∧(¬D∨G) .(A∨B)∧(¬A∨C)∧(¬B∨D)∧(¬C∨G)∧(¬D∨G) .

1. Prove using resolution that the above sentence entails GG.  
2. Two clauses are *semantically distinct* if they are not logically equivalent. How many semantically distinct 2-CNF clauses can be constructed from nn proposition symbols?  
3. Using your answer to (b), prove that propositional resolution always terminates in time polynomial in nn given a 2-CNF sentence containing no more than nn distinct symbols.  
4. Explain why your argument in (c) does not apply to 3-CNF.

[Exercise 22](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_22/)

Prove each of the following assertions:  
1. Every pair of propositional clauses either has no resolvents, or all their resolvents are logically equivalent.  
2. There is no clause that, when resolved with itself, yields (after factoring) the clause (¬P∨¬Q)(¬P∨¬Q).  
3. If a propositional clause CC can be resolved with a copy of itself, it must be logically equivalent to TrueTrue.

[Exercise 23](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_23/)

Consider the following sentence:

[(Food⇒Party)∨(Drinks⇒Party)]⇒[(Food∧Drinks)⇒Party] .[(Food⇒Party)∨(Drinks⇒Party)]⇒[(Food∧Drinks)⇒Party] .

1. Determine, using enumeration, whether this sentence is valid, satisfiable (but not valid), or unsatisfiable.  
2. Convert the left-hand and right-hand sides of the main implication into CNF, showing each step, and explain how the results confirm your answer to (a).  
3. Prove your answer to (a) using resolution.

[Exercise 24 (dnf-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_24/)

A sentence is in disjunctive normal form(DNF) if it is the disjunction of conjunctions of literals. For example, the sentence (A∧B∧¬C)∨(¬A∧C)∨(B∧¬C)(A∧B∧¬C)∨(¬A∧C)∨(B∧¬C) is in DNF.  
1. Any propositional logic sentence is logically equivalent to the assertion that some possible world in which it would be true is in fact the case. From this observation, prove that any sentence can be written in DNF.  
2. Construct an algorithm that converts any sentence in propositional logic into DNF. (*Hint*: The algorithm is similar to the algorithm for conversion to CNF iven in Sectio .)  
3. Construct a simple algorithm that takes as input a sentence in DNF and returns a satisfying assignment if one exists, or reports that no satisfying assignment exists.  
4. Apply the algorithms in (b) and (c) to the following set of sentences:  
A⇒BA⇒B  
B⇒CB⇒C  
C⇒AC⇒A  
5. Since the algorithm in (b) is very similar to the algorithm for conversion to CNF, and since the algorithm in (c) is much simpler than any algorithm for solving a set of sentences in CNF, why is this technique not used in automated reasoning?

[Exercise 25 (convert-clausal-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_25/)

Convert the following set of sentences to clausal form.  
1. S1: A⇔(B∨E)A⇔(B∨E).  
2. S2: E⇒DE⇒D.  
3. S3: C∧F⇒¬BC∧F⇒¬B.  
4. S4: E⇒BE⇒B.  
5. S5: B⇒FB⇒F.  
6. S6: B⇒CB⇒C  
Give a trace of the execution of DPLL on the conjunction of these clauses.

[Exercise 26 (convert-clausal-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_2/)

Convert the following set of sentences to clausal form.  
1. S1: A⇔(B∨E)A⇔(B∨E).  
2. S2: E⇒DE⇒D.  
3. S3: C∧F⇒¬BC∧F⇒¬B.  
4. S4: E⇒BE⇒B.  
5. S5: B⇒FB⇒F.  
6. S6: B⇒CB⇒C  
Give a trace of the execution of DPLL on the conjunction of these clauses.

[Exercise 27](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_27/)

Is a randomly generated 4-CNF sentence with nn symbols and mm clauses more or less likely to be solvable than a randomly generated 3-CNF sentence with nn symbols and mm clauses? Explain.

[Exercise 28 (minesweeper-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_28/)

Minesweeper, the well-known computer game, is closely related to the wumpus world. A minesweeper world is a rectangular grid of NN squares with MM invisible mines scattered among them. Any square may be probed by the agent; instant death follows if a mine is probed. Minesweeper indicates the presence of mines by revealing, in each probed square, the *number* of mines that are directly or diagonally adjacent. The goal is to probe every unmined square. 1. Let Xi,jXi,j be true iff square [i,j][i,j] contains a mine. Write down the assertion that exactly two mines are adjacent to \[1,1\] as a sentence involving some logical combination of Xi,jXi,j propositions. 2. Generalize your assertion from (a) by explaining how to construct a CNF sentence asserting that kk of nn neighbors contain mines. 3. Explain precisely how an agent can use {DPLL} to prove that a given square does (or does not) contain a mine, ignoring the global constraint that there are exactly MM mines in all. 4. Suppose that the global constraint is constructed from your method from part (b). How does the number of clauses depend on MM and NN? Suggest a way to modify {DPLL} so that the global constraint does not need to be represented explicitly. 5. Are any conclusions derived by the method in part (c) invalidated when the global constraint is taken into account? 6. Give examples of configurations of probe values that induce *long-range dependencies* such that the contents of a given unprobed square would give information about the contents of a far-distant square. (*Hint*: consider an N×1N×1 board.)

[Exercise 29 (known-literal-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_29/)

How long does it take to prove KB⊨αKB⊨α using {DPLL} when αα is a literal *already contained in* KBKB? Explain.

[Exercise 30 (dpll-fc-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_30/)

Trace the behavior of {DPLL} on the knowledge base in Figure [7.16](https://aimacode.github.io/aima-exercises/figures/pl-horn-example-figure.png) when trying to prove QQ, and compare this behavior with that of the forward-chaining algorithm.

[Exercise 31](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_31/)

Write a successor-state axiom for the LockedLocked predicate, which applies to doors, assuming the only actions available are LockLock and UnlockUnlock.

[Exercise 32](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_32/)

Discuss what is meant by *optimal* behavior in the wumpus world. Show that the {Hybrid-Wumpus-Agent} is not optimal, and suggest ways to improve it.

[Exercise 33](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_3/)

Suppose an agent inhabits a world with two states, SS and ¬S¬S, and can do exactly one of two actions, aa and bb. Action aa does nothing and action bb flips from one state to the other. Let StSt be the proposition that the agent is in state SS at time tt, and let atat be the proposition that the agent does action aa at time tt (similarly for btbt).  
1. Write a successor-state axiom for St+1St+1.  
2. Convert the sentence in (a) into CNF.  
3. Show a resolution refutation proof that if the agent is in ¬S¬S at time tt and does aa, it will still be in ¬S¬S at time t+1t+1.

[Exercise 34 (ss-axiom-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_34/)

Section  provides some of the successor-state axioms required for the wumpus world. Write down axioms for all remaining fluent symbols.

[Exercise 35 (hybrid-wumpus-exercise)](https://aimacode.github.io/aima-exercises/knowledge-logic-exercises/ex_35/)

Modify the {Hybrid-Wumpus-Agent} to use the 1-CNF logical state estimation method described on page . We noted on that page that such an agent will not be able to acquire, maintain, and use more complex beliefs such as the disjunction P3,1∨P2,2P3,1∨P2,2. Suggest a method for overcoming this problem by defining additional proposition symbols, and try it out in the wumpus world. Does it improve the performance of the agent?

8. First Order Logic

[Exercise 1](https://aimacode.github.io/aima-exercises/fol-exercises/ex_1/)

A logical knowledge base represents the world using a set of sentences with no explicit structure. An **analogical** representation, on the other hand, has physical structure that corresponds directly to the structure of the thing represented. Consider a road map of your country as an analogical representation of facts about the country—it represents facts with a map language. The two-dimensional structure of the map corresponds to the two-dimensional surface of the area.  
1. Give five examples of *symbols* in the map language.  
2. An *explicit* sentence is a sentence that the creator of the representation actually writes down. An *implicit* sentence is a sentence that results from explicit sentences because of properties of the analogical representation. Give three examples each of *implicit* and *explicit* sentences in the map language.  
3. Give three examples of facts about the physical structure of your country that cannot be represented in the map language.  
4. Give two examples of facts that are much easier to express in the map language than in first-order logic.  
5. Give two other examples of useful analogical representations. What are the advantages and disadvantages of each of these languages?

[Exercise 2](https://aimacode.github.io/aima-exercises/fol-exercises/ex_2/)

Consider a knowledge base containing just two sentences: P(a)P(a) and P(b)P(b). Does this knowledge base entail ∀x P(x)∀x P(x)? Explain your answer in terms of models.

[Exercise 3](https://aimacode.github.io/aima-exercises/fol-exercises/ex_3/)

Is the sentence ∃x,yxy∃x,yxy valid? Explain.

[Exercise 4](https://aimacode.github.io/aima-exercises/fol-exercises/ex_4/)

Write down a logical sentence such that every world in which it is true contains exactly one object.

[Exercise 5 (two-friends-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_5/)

Write down a logical sentence such that every world in which it is true contains exactly two objects.

[Exercise 6 (fol-model-count-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_6/)

Consider a symbol vocabulary that contains cc constant symbols, pkpk predicate symbols of each arity kk, and fkfk function symbols of each arity kk, where 1≤k≤A1≤k≤A. Let the domain size be fixed at DD. For any given model, each predicate or function symbol is mapped onto a relation or function, respectively, of the same arity. You may assume that the functions in the model allow some input tuples to have no value for the function (i.e., the value is the invisible object). Derive a formula for the number of possible models for a domain with DD elements. Don’t worry about eliminating redundant combinations.

[Exercise 7 (nqueens-size-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_7/)

Which of the following are valid (necessarily true) sentences?  
1. (∃x xx)⇒(∀y∃z yz)(∃x xx)⇒(∀y∃z yz).  
2. ∀xP(x)∨¬P(x)∀xP(x)∨¬P(x).  
3. ∀xSmart(x)∨(xx)∀xSmart(x)∨(xx).

[Exercise 8 (empty-universe-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_8/)

Consider a version of the semantics for first-order logic in which models with empty domains are allowed. Give at least two examples of sentences that are valid according to the standard semantics but not according to the new semantics. Discuss which outcome makes more intuitive sense for your examples.

[Exercise 9 (hillary-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_9/)

Does the fact ¬Spouse(George,Laura)¬Spouse(George,Laura) follow from the facts Jim≠GeorgeJim≠George and Spouse(Jim,Laura)Spouse(Jim,Laura)? If so, give a proof; if not, supply additional axioms as needed. What happens if we use SpouseSpouse as a unary function symbol instead of a binary predicate?

[Exercise 10](https://aimacode.github.io/aima-exercises/fol-exercises/ex_10/)

This exercise uses the function MapColorMapColor and predicates In(x,y)In(x,y), Borders(x,y)Borders(x,y), and Country(x)Country(x), whose arguments are geographical regions, along with constant symbols for various regions. In each of the following we give an English sentence and a number of candidate logical expressions. For each of the logical expressions, state whether it (1) correctly expresses the English sentence; (2) is syntactically invalid and therefore meaningless; or (3) is syntactically valid but does not express the meaning of the English sentence.  
1. Paris and Marseilles are both in France.  
1. In(Paris∧Marseilles,France)In(Paris∧Marseilles,France).  
2. In(Paris,France)∧In(Marseilles,France)In(Paris,France)∧In(Marseilles,France).  
3. In(Paris,France)∨In(Marseilles,France)In(Paris,France)∨In(Marseilles,France).  
2. There is a country that borders both Iraq and Pakistan.  
1. ∃c∃c Country(c)∧Border(c,Iraq)∧Border(c,Pakistan)Country(c)∧Border(c,Iraq)∧Border(c,Pakistan).  
2. ∃c∃c Country(c)⇒[Border(c,Iraq)∧Border(c,Pakistan)]Country(c)⇒[Border(c,Iraq)∧Border(c,Pakistan)].  
3. [∃c[∃c Country(c)]⇒[Border(c,Iraq)∧Border(c,Pakistan)]Country(c)]⇒[Border(c,Iraq)∧Border(c,Pakistan)].  
4. ∃c∃c Border(Country(c),Iraq∧Pakistan)Border(Country(c),Iraq∧Pakistan).  
3. All countries that border Ecuador are in South America.  
1. ∀cCountry(c)∧Border(c,Ecuador)⇒In(c,SouthAmerica)∀cCountry(c)∧Border(c,Ecuador)⇒In(c,SouthAmerica).  
2. ∀cCountry(c)⇒[Border(c,Ecuador)⇒In(c,SouthAmerica)]∀cCountry(c)⇒[Border(c,Ecuador)⇒In(c,SouthAmerica)].  
3. ∀c[Country(c)⇒Border(c,Ecuador)]⇒In(c,SouthAmerica)∀c[Country(c)⇒Border(c,Ecuador)]⇒In(c,SouthAmerica).  
4. ∀cCountry(c)∧Border(c,Ecuador)∧In(c,SouthAmerica)∀cCountry(c)∧Border(c,Ecuador)∧In(c,SouthAmerica).  
4. No region in South America borders any region in Europe.  
1. ¬[∃c,dIn(c,SouthAmerica)∧In(d,Europe)∧Borders(c,d)]¬[∃c,dIn(c,SouthAmerica)∧In(d,Europe)∧Borders(c,d)].  
2. ∀c,d[In(c,SouthAmerica)∧In(d,Europe)]⇒¬Borders(c,d)]∀c,d[In(c,SouthAmerica)∧In(d,Europe)]⇒¬Borders(c,d)].  
3. ¬∀cIn(c,SouthAmerica)⇒∃dIn(d,Europe)∧¬Borders(c,d)¬∀cIn(c,SouthAmerica)⇒∃dIn(d,Europe)∧¬Borders(c,d). 4. ∀cIn(c,SouthAmerica)⇒∀dIn(d,Europe)⇒¬Borders(c,d)∀cIn(c,SouthAmerica)⇒∀dIn(d,Europe)⇒¬Borders(c,d).  
5. No two adjacent countries have the same map color.  
1. ∀x,y¬Country(x)∨¬Country(y)∨¬Borders(x,y)∨∀x,y¬Country(x)∨¬Country(y)∨¬Borders(x,y)∨\ ¬(MapColor(x)=MapColor(y))¬(MapColor(x)=MapColor(y)).  
2. ∀x,y(Country(x)∧Country(y)∧Borders(x,y)∧¬(x=y))⇒∀x,y(Country(x)∧Country(y)∧Borders(x,y)∧¬(x=y))⇒\ ¬(MapColor(x)=MapColor(y))¬(MapColor(x)=MapColor(y)).  
3. ∀x,yCountry(x)∧Country(y)∧Borders(x,y)∧∀x,yCountry(x)∧Country(y)∧Borders(x,y)∧\ ¬(MapColor(x)=MapColor(y))¬(MapColor(x)=MapColor(y)).  
4. ∀x,y(Country(x)∧Country(y)∧Borders(x,y))⇒MapColor(x≠y)∀x,y(Country(x)∧Country(y)∧Borders(x,y))⇒MapColor(x≠y).

[Exercise 11](https://aimacode.github.io/aima-exercises/fol-exercises/ex_11/)

Consider a vocabulary with the following symbols:  
> Occupation(p,o)Occupation(p,o): Predicate. Person pp has occupation oo. > Customer(p1,p2)Customer(p1,p2): Predicate. Person p1p1 is a customer of person p2p2. > Boss(p1,p2)Boss(p1,p2): Predicate. Person p1p1 is a boss of person p2p2. > DoctorDoctor, SurgeonSurgeon, LawyerLawyer, ActorActor: Constants denoting occupations. > EmilyEmily, JoeJoe: Constants denoting people. Use these symbols to write the following assertions in first-order logic:  
1. Emily is either a surgeon or a lawyer.  
2. Joe is an actor, but he also holds another job.  
3. All surgeons are doctors.  
4. Joe does not have a lawyer (i.e., is not a customer of any lawyer).  
5. Emily has a boss who is a lawyer.  
6. There exists a lawyer all of whose customers are doctors.  
7. Every surgeon has a lawyer.

[Exercise 12](https://aimacode.github.io/aima-exercises/fol-exercises/ex_12/)

In each of the following we give an English sentence and a number of candidate logical expressions. For each of the logical expressions, state whether it (1) correctly expresses the English sentence; (2) is syntactically invalid and therefore meaningless; or (3) is syntactically valid but does not express the meaning of the English sentence.  
1. Every cat loves its mother or father.  
1. ∀xCat(x)⇒Loves(x,Mother(x)∨Father(x))∀xCat(x)⇒Loves(x,Mother(x)∨Father(x)).  
2. ∀x¬Cat(x)∨Loves(x,Mother(x))∨Loves(x,Father(x))∀x¬Cat(x)∨Loves(x,Mother(x))∨Loves(x,Father(x)).  
3. ∀xCat(x)∧(Loves(x,Mother(x))∨Loves(x,Father(x)))∀xCat(x)∧(Loves(x,Mother(x))∨Loves(x,Father(x))).  
2. Every dog who loves one of its brothers is happy.  
1. ∀xDog(x)∧(∃y Brother(y,x)∧Loves(x,y))⇒Happy(x)∀xDog(x)∧(∃y Brother(y,x)∧Loves(x,y))⇒Happy(x).  
2. ∀x,yDog(x)∧Brother(y,x)∧Loves(x,y)⇒Happy(x)∀x,yDog(x)∧Brother(y,x)∧Loves(x,y)⇒Happy(x).  
3. ∀xDog(x)∧[∀yBrother(y,x)⇔Loves(x,y)]⇒Happy(x)∀xDog(x)∧[∀yBrother(y,x)⇔Loves(x,y)]⇒Happy(x).  
3. No dog bites a child of its owner.  
1. ∀xDog(x)⇒¬Bites(x,Child(Owner(x)))∀xDog(x)⇒¬Bites(x,Child(Owner(x))).  
2. ¬∃x,yDog(x)∧Child(y,Owner(x))∧Bites(x,y)¬∃x,yDog(x)∧Child(y,Owner(x))∧Bites(x,y).  
3. ∀xDog(x)⇒(∀yChild(y,Owner(x))⇒¬Bites(x,y))∀xDog(x)⇒(∀yChild(y,Owner(x))⇒¬Bites(x,y)).  
4. ¬∃xDog(x)⇒(∃yChild(y,Owner(x))∧Bites(x,y))¬∃xDog(x)⇒(∃yChild(y,Owner(x))∧Bites(x,y)).  
4. Everyone’s zip code within a state has the same first digit.  
1. ∀x,s,z1[State(s)∧LivesIn(x,s)∧Zip(x)z1]⇒∀x,s,z1[State(s)∧LivesIn(x,s)∧Zip(x)z1]⇒\ [∀y,z2LivesIn(y,s)∧Zip(y)z2⇒Digit(1,z1)Digit(1,z2)][∀y,z2LivesIn(y,s)∧Zip(y)z2⇒Digit(1,z1)Digit(1,z2)].  
2. ∀x,s[State(s)∧LivesIn(x,s)∧∃z1Zip(x)z1]⇒∀x,s[State(s)∧LivesIn(x,s)∧∃z1Zip(x)z1]⇒\ [∀y,z2LivesIn(y,s)∧Zip(y)z2∧Digit(1,z1)Digit(1,z2)][∀y,z2LivesIn(y,s)∧Zip(y)z2∧Digit(1,z1)Digit(1,z2)].  
3. ∀x,y,sState(s)∧LivesIn(x,s)∧LivesIn(y,s)⇒Digit(1,Zip(x)Zip(y))∀x,y,sState(s)∧LivesIn(x,s)∧LivesIn(y,s)⇒Digit(1,Zip(x)Zip(y)).  
4. ∀x,y,sState(s)∧LivesIn(x,s)∧LivesIn(y,s)⇒∀x,y,sState(s)∧LivesIn(x,s)∧LivesIn(y,s)⇒\ Digit(1,Zip(x))Digit(1,Zip(y))Digit(1,Zip(x))Digit(1,Zip(y)).

[Exercise 13 (language-determination-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_13/)

Complete the following exercises about logical sentences:  
1. Translate into \*good, natural\* English (no xxs or yys!):

∀x,y,lSpeaksLanguage(x,l)∧SpeaksLanguage(y,l)⟹Understands(x,y)∧Understands(y,x).∀x,y,lSpeaksLanguage(x,l)∧SpeaksLanguage(y,l)⟹Understands(x,y)∧Understands(y,x).

2. Explain why this sentence is entailed by the sentence

∀x,y,lSpeaksLanguage(x,l)∧SpeaksLanguage(y,l)⟹Understands(x,y).∀x,y,lSpeaksLanguage(x,l)∧SpeaksLanguage(y,l)⟹Understands(x,y).

3. Translate into first-order logic the following sentences:  
1. Understanding leads to friendship.  
2. Friendship is transitive.  
Remember to define all predicates, functions, and constants you use.

[Exercise 14](https://aimacode.github.io/aima-exercises/fol-exercises/ex_14/)

True or false? Explain.  
1. ∃xxRumpelstiltskin∃xxRumpelstiltskin is a valid (necessarily true) sentence of first-order logic.  
2. Every existentially quantified sentence in first-order logic is true in any model that contains exactly one object.  
3. ∀x,yxy∀x,yxyis satisfiable.

[Exercise 15 (Peano-completion-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_15/)

Rewrite the first two Peano axioms in Section  as a single axiom that defines NatNum(x)NatNum(x) so as to exclude the possibility of natural numbers except for those generated by the successor function.

[Exercise 16 (wumpus-diagnostic-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_16/)

Equation () on page  defines the conditions under which a square is breezy. Here we consider two other ways to describe this aspect of the wumpus world.  
1. We can write [diagnostic rule] leading from observed effects to hidden causes. For finding pits, the obvious diagnostic rules say that if a square is breezy, some adjacent square must contain a pit; and if a square is not breezy, then no adjacent square contains a pit. Write these two rules in first-order logic and show that their conjunction is logically equivalent to Equation ([pit-biconditional-equation](https://aimacode.github.io/aima-exercises/fol-exercises/)).  
2. We can write [causal rule] leading from cause to effect. One obvious causal rule is that a pit causes all adjacent squares to be breezy. Write this rule in first-order logic, explain why it is incomplete compared to Equation ([pit-biconditional-equation](https://aimacode.github.io/aima-exercises/fol-exercises/)), and supply the missing axiom.

[Exercise 17 (kinship-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_17/)

Write axioms describing the predicates GrandchildGrandchild, GreatgrandparentGreatgrandparent, AncestorAncestor, BrotherBrother, SisterSister, DaughterDaughter, SonSon, FirstCousinFirstCousin, BrotherInLawBrotherInLaw, SisterInLawSisterInLaw, AuntAunt, and UncleUncle. Find out the proper definition of mmth cousin nn times removed, and write the definition in first-order logic. Now write down the basic facts depicted in the family tree in Figure . Using a suitable logical reasoning system, it all the sentences you have written down, and it who are Elizabeth’s grandchildren, Diana’s brothers-in-law, Zara’s great-grandparents, and Eugenie’s ancestors.

**A typical family tree. The symbol**⋈⋈**connects spouses and arrows point to children.**

[Exercise 18](https://aimacode.github.io/aima-exercises/fol-exercises/ex_18/)

Write down a sentence asserting that + is a commutative function. Does your sentence follow from the Peano axioms? If so, explain why; if not, give a model in which the axioms are true and your sentence is false.

[Exercise 19](https://aimacode.github.io/aima-exercises/fol-exercises/ex_19/)

Explain what is wrong with the following proposed definition of the set membership predicate

∀,x,s;;x∈x|s∀,x,s;;x∈x|s

∀,x,s;;x∈s⟹∀,y;;x∈y|s∀,x,s;;x∈s⟹∀,y;;x∈y|s

[Exercise 20 (list-representation-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_20/)

Using the set axioms as examples, write axioms for the list domain, including all the constants, functions, and predicates mentioned in the chapter.

[Exercise 21 (adjacency-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_21/)

Explain what is wrong with the following proposed definition of adjacent squares in the wumpus world:

∀x,yAdjacent([x,y],[x+1,y])∧Adjacent([x,y],[x,y+1]) .∀x,yAdjacent([x,y],[x+1,y])∧Adjacent([x,y],[x,y+1]) .

[Exercise 22](https://aimacode.github.io/aima-exercises/fol-exercises/ex_22/)

Write out the axioms required for reasoning about the wumpus’s location, using a constant symbol WumpusWumpus and a binary predicate At(Wumpus,Location)At(Wumpus,Location). Remember that there is only one wumpus.

[Exercise 23](https://aimacode.github.io/aima-exercises/fol-exercises/ex_23/)

Assuming predicates Parent(p,q)Parent(p,q) and Female(p)Female(p) and constants JoanJoan and KevinKevin, with the obvious meanings, express each of the following sentences in first-order logic. (You may use the abbreviation ∃1∃1 to mean “there exists exactly one.”)  
1. Joan has a daughter (possibly more than one, and possibly sons as well).  
2. Joan has exactly one daughter (but may have sons as well).  
3. Joan has exactly one child, a daughter.  
4. Joan and Kevin have exactly one child together.  
5. Joan has at least one child with Kevin, and no children with anyone else.

[Exercise 24](https://aimacode.github.io/aima-exercises/fol-exercises/ex_24/)

Arithmetic assertions can be written in first-order logic with the predicate symbol <<, the function symbols ++ and ××, and the constant symbols 0 and 1. Additional predicates can also be defined with biconditionals.  
1. Represent the property “xx is an even number.”  
2. Represent the property “xx is prime.”  
3. Goldbach’s conjecture is the conjecture (unproven as yet) that every even number is equal to the sum of two primes. Represent this conjecture as a logical sentence.

[Exercise 25](https://aimacode.github.io/aima-exercises/fol-exercises/ex_25/)

In Chapter [Constraint Satisfaction Problems](https://aimacode.github.io/aima-exercises/csp-exercises/), we used equality to indicate the relation between a variable and its value. For instance, we wrote WAredWAred to mean that Western Australia is colored red. Representing this in first-order logic, we must write more verbosely ColorOf(WA)redColorOf(WA)red. What incorrect inference could be drawn if we wrote sentences such as WAredWAred directly as logical assertions?

[Exercise 26](https://aimacode.github.io/aima-exercises/fol-exercises/ex_26/)

Write in first-order logic the assertion that every key and at least one of every pair of socks will eventually be lost forever, using only the following vocabulary: Key(x)Key(x), xx is a key; Sock(x)Sock(x), xx is a sock; Pair(x,y)Pair(x,y), xx and yy are a pair; NowNow, the current time; Before(t1,t2)Before(t1,t2), time t1t1 comes before time t2t2; Lost(x,t)Lost(x,t), object xx is lost at time tt.

[Exercise 27](https://aimacode.github.io/aima-exercises/fol-exercises/ex_27/)

For each of the following sentences in English, decide if the accompanying first-order logic sentence is a good translation. If not, explain why not and correct it. (Some sentences may have more than one error!)  
1. No two people have the same social security number.

¬∃x,y,nPerson(x)∧Person(y)⇒[HasSS#(x,n)∧HasSS#(y,n)].¬∃x,y,nPerson(x)∧Person(y)⇒[HasSS#(x,n)∧HasSS#(y,n)].

2. John’s social security number is the same as Mary’s.

∃nHasSS#(John,n)∧HasSS#(Mary,n).∃nHasSS#(John,n)∧HasSS#(Mary,n).

3. Everyone’s social security number has nine digits.

∀x,nPerson(x)⇒[HasSS#(x,n)∧Digits(n,9)].∀x,nPerson(x)⇒[HasSS#(x,n)∧Digits(n,9)].

4. Rewrite each of the above (uncorrected) sentences using a function symbol SS#SS# instead of the predicate HasSS#HasSS#.

[Exercise 28](https://aimacode.github.io/aima-exercises/fol-exercises/ex_28/)

Translate into first-order logic the sentence “Everyone’s DNA is unique and is derived from their parents’ DNA.” You must specify the precise intended meaning of your vocabulary terms. (\*Hint\*: Do not use the predicate Unique(x)Unique(x), since uniqueness is not really a property of an object in itself!)

[Exercise 29](https://aimacode.github.io/aima-exercises/fol-exercises/ex_29/)

For each of the following sentences in English, decide if the accompanying first-order logic sentence is a good translation. If not, explain why not and correct it.  
1. Any apartment in London has lower rent than some apartments in Paris.

∀x[Apt(x)∧In(x,London)]⟹∃y([Apt(y)∧In(y,Paris)]⟹(Rent(x)<Rent(y)))∀x[Apt(x)∧In(x,London)]⟹∃y([Apt(y)∧In(y,Paris)]⟹(Rent(x)<Rent(y)))

2. There is exactly one apartment in Paris with rent below $1000.

∃xApt(x)∧In(x,Paris)∧∀y[Apt(y)∧In(y,Paris)∧(Rent(y)<Dollars(1000))]⟹(y=x)∃xApt(x)∧In(x,Paris)∧∀y[Apt(y)∧In(y,Paris)∧(Rent(y)<Dollars(1000))]⟹(y=x)

3. If an apartment is more expensive than all apartments in London, it must be in Moscow.

∀xApt(x)∧[∀yApt(y)∧In(y,London)∧(Rent(x)>Rent(y))]⟹In(x,Moscow).∀xApt(x)∧[∀yApt(y)∧In(y,London)∧(Rent(x)>Rent(y))]⟹In(x,Moscow).

[Exercise 30](https://aimacode.github.io/aima-exercises/fol-exercises/ex_30/)

Represent the following sentences in first-order logic, using a consistent vocabulary (which you must define):  
1. Some students took French in spring 2001.  
2. Every student who takes French passes it.  
3. Only one student took Greek in spring 2001.  
4. The best score in Greek is always higher than the best score in French.  
5. Every person who buys a policy is smart.  
6. No person buys an expensive policy.  
7. There is an agent who sells policies only to people who are not insured.  
8. There is a barber who shaves all men in town who do not shave themselves.  
9. A person born in the UK, each of whose parents is a UK citizen or a UK resident, is a UK citizen by birth.  
10. A person born outside the UK, one of whose parents is a UK citizen by birth, is a UK citizen by descent.  
11. Politicians can fool some of the people all of the time, and they can fool all of the people some of the time, but they can’t fool all of the people all of the time.  
12. All Greeks speak the same language. (Use Speaks(x,l)Speaks(x,l) to mean that person xx speaks language ll.)

[Exercise 31](https://aimacode.github.io/aima-exercises/fol-exercises/ex_31/)

Represent the following sentences in first-order logic, using a consistent vocabulary (which you must define):  
1. Some students took French in spring 2001.  
2. Every student who takes French passes it.  
3. Only one student took Greek in spring 2001.  
4. The best score in Greek is always higher than the best score in French.  
5. Every person who buys a policy is smart.  
6. No person buys an expensive policy.  
7. There is an agent who sells policies only to people who are not insured.  
8. There is a barber who shaves all men in town who do not shave themselves.  
9. A person born in the UK, each of whose parents is a UK citizen or a UK resident, is a UK citizen by birth.  
10. A person born outside the UK, one of whose parents is a UK citizen by birth, is a UK citizen by descent.  
11. Politicians can fool some of the people all of the time, and they can fool all of the people some of the time, but they can’t fool all of the people all of the time.  
12. All Greeks speak the same language. (Use Speaks(x,l)Speaks(x,l) to mean that person xx speaks language ll.)

[Exercise 32](https://aimacode.github.io/aima-exercises/fol-exercises/ex_32/)

Write a general set of facts and axioms to represent the assertion “Wellington heard about Napoleon’s death” and to correctly answer the question “Did Napoleon hear about Wellington’s death?”

[Exercise 33 (4bit-adder-exercise)](https://aimacode.github.io/aima-exercises/fol-exercises/ex_33/)

Extend the vocabulary from Section  to define addition for nn-bit binary numbers. Then encode the description of the four-bit adder in Figure [4bit-adder-figure](https://aimacode.github.io/aima-exercises/fol-exercises/#4bit-adder-figure), and pose the queries needed to verify that it is in fact correct.

**A four-bit adder. Each**AdiAdi**is a one-bit adder, as in figure**[**adder-figure**](https://aimacode.github.io/aima-exercises/fol-exercises/#4bit-adder-figure)**on page <a href=""#">adder-figure</a>**

[Exercise 34](https://aimacode.github.io/aima-exercises/fol-exercises/ex_34/)

The circuit representation in the chapter is more detailed than necessary if we care only about circuit functionality. A simpler formulation describes any mm-input, nn-output gate or circuit using a predicate with m+nm+n arguments, such that the predicate is true exactly when the inputs and outputs are consistent. For example, NOT gates are described by the binary predicate NOT(i,o)NOT(i,o), for which NOT(0,1)NOT(0,1) and NOT(1,0)NOT(1,0) are known. Compositions of gates are defined by conjunctions of gate predicates in which shared variables indicate direct connections. For example, a NAND circuit can be composed from ANDANDs and NOTNOTs:

∀i1,i2,oa,oAND(i1,i2,oa)∧NOT(oa,o)⇒NAND(i1,i2,o) .∀i1,i2,oa,oAND(i1,i2,oa)∧NOT(oa,o)⇒NAND(i1,i2,o) .

Using this representation, define the one-bit adder in Figure [adder-figure](https://aimacode.github.io/aima-exercises/fol-exercises/#4bit-adder-figure) and the four-bit adder in Figure [adder-figure](https://aimacode.github.io/aima-exercises/fol-exercises/#4bit-adder-figure), and explain what queries you would use to verify the designs. What kinds of queries are \*not\* supported by this representation that \*are\* supported by the representation in Section ?

[Exercise 35](https://aimacode.github.io/aima-exercises/fol-exercises/ex_35/)

Obtain a passport application for your country, identify the rules determining eligibility for a passport, and translate them into first-order logic, following the steps outlined in Section

[Exercise 36](https://aimacode.github.io/aima-exercises/fol-exercises/ex_36/)

Consider a first-order logical knowledge base that describes worlds containing people, songs, albums (e.g., “Meet the Beatles”) and disks (i.e., particular physical instances of CDs). The vocabulary contains the following symbols:  
> CopyOf(d,a)CopyOf(d,a): Predicate. Disk dd is a copy of album aa. > Owns(p,d)Owns(p,d): Predicate. Person pp owns disk dd. > Sings(p,s,a)Sings(p,s,a): Album aa includes a recording of song ss sung by person pp. > Wrote(p,s)Wrote(p,s): Person pp wrote song ss. > McCartneyMcCartney, GershwinGershwin, BHolidayBHoliday, JoeJoe, EleanorRigbyEleanorRigby, TheManILoveTheManILove, RevolverRevolver: Constants with the obvious meanings. Express the following statements in first-order logic:  
1. Gershwin wrote “The Man I Love.”  
2. Gershwin did not write “Eleanor Rigby.”  
3. Either Gershwin or McCartney wrote “The Man I Love.”  
4. Joe has written at least one song.  
5. Joe owns a copy of \*Revolver\*.  
6. Every song that McCartney sings on \*Revolver\* was written by McCartney.  
7. Gershwin did not write any of the songs on \*Revolver\*.  
8. Every song that Gershwin wrote has been recorded on some album. (Possibly different songs are recorded on different albums.)  
9. There is a single album that contains every song that Joe has written.  
10. Joe owns a copy of an album that has Billie Holiday singing “The Man I Love.”  
11. Joe owns a copy of every album that has a song sung by McCartney. (Of course, each different album is instantiated in a different physical CD.)  
12. Joe owns a copy of every album on which all the songs are sung by Billie Holiday.

9. Inference in First-Order Logic

[Exercise 1](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_1/)

Prove that Universal Instantiation is sound and that Existential Instantiation produces an inferentially equivalent knowledge base.

[Exercise 2](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_2/)

From Likes(Jerry,IceCream)Likes(Jerry,IceCream) it seems reasonable to infer ∃xLikes(x,IceCream)∃xLikes(x,IceCream). Write down a general inference rule, , that sanctions this inference. State carefully the conditions that must be satisfied by the variables and terms involved.

[Exercise 3](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_3/)

Suppose a knowledge base contains just one sentence, ∃x AsHighAs(x,Everest)∃x AsHighAs(x,Everest). Which of the following are legitimate results of applying Existential Instantiation?  
1. AsHighAs(Everest,Everest)AsHighAs(Everest,Everest).  
2. AsHighAs(Kilimanjaro,Everest)AsHighAs(Kilimanjaro,Everest).  
3. AsHighAs(Kilimanjaro,Everest)∧AsHighAs(BenNevis,Everest)AsHighAs(Kilimanjaro,Everest)∧AsHighAs(BenNevis,Everest)\ (after two applications).

[Exercise 4](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_4/)

For each pair of atomic sentences, give the most general unifier if it exists:  
1. P(A,B,B)P(A,B,B), P(x,y,z)P(x,y,z).  
2. Q(y,G(A,B))Q(y,G(A,B)), Q(G(x,x),y)Q(G(x,x),y).  
3. Older(Father(y),y)Older(Father(y),y), Older(Father(x),John)Older(Father(x),John).  
4. Knows(Father(y),y)Knows(Father(y),y), Knows(x,x)Knows(x,x).

[Exercise 5](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_5/)

For each pair of atomic sentences, give the most general unifier if it exists:  
1. P(A,B,B)P(A,B,B), P(x,y,z)P(x,y,z).  
2. Q(y,G(A,B))Q(y,G(A,B)), Q(G(x,x),y)Q(G(x,x),y).  
3. Older(Father(y),y)Older(Father(y),y), Older(Father(x),John)Older(Father(x),John).  
4. Knows(Father(y),y)Knows(Father(y),y), Knows(x,x)Knows(x,x).

[Exercise 6 (subsumption-lattice-exercise)](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_6/)

Consider the subsumption lattices shown in Figure [9.2](https://aimacode.github.io/aima-exercises/figures/subsumption-lattice-figure.png) (page   
. 1. Construct the lattice for the sentence Employs(Mother(John),Father(Richard))Employs(Mother(John),Father(Richard)).  
2. Construct the lattice for the sentence Employs(IBM,y)Employs(IBM,y) (“Everyone works for IBM”). Remember to include every kind of query that unifies with the sentence.  
3. Assume that indexes each sentence under every node in its subsumption lattice. Explain how should work when some of these sentences contain variables; use as examples the sentences in (a) and (b) and the query Employs(x,Father(x))Employs(x,Father(x)).

[Exercise 7 (fol-horses-exercise)](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_7/)

Write down logical representations for the following sentences, suitable for use with Generalized Modus Ponens:  
1. Horses, cows, and pigs are mammals.  
2. An offspring of a horse is a horse.  
3. Bluebeard is a horse.  
4. Bluebeard is Charlie’s parent.  
5. Offspring and parent are inverse relations.  
6. Every mammal has a parent.

[Exercise 8](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_8/)

These questions concern concern issues with substitution and Skolemization.  
1. Given the premise ∀x∃yP(x,y)∀x∃yP(x,y), it is not valid to conclude that ∃qP(q,q)∃qP(q,q). Give an example of a predicate PP where the first is true but the second is false.  
2. Suppose that an inference engine is incorrectly written with the occurs check omitted, so that it allows a literal like P(x,F(x))P(x,F(x)) to be unified with P(q,q)P(q,q). (As mentioned, most standard implementations of Prolog actually do allow this.) Show that such an inference engine will allow the conclusion ∃yP(q,q)∃yP(q,q) to be inferred from the premise ∀x∃yP(x,y)∀x∃yP(x,y).  
3. Suppose that a procedure that converts first-order logic to clausal form incorrectly Skolemizes ∀x∃yP(x,y)∀x∃yP(x,y) to P(x,Sk0)P(x,Sk0)—that is, it replaces yy by a Skolem constant rather than by a Skolem function of xx. Show that an inference engine that uses such a procedure will likewise allow ∃qP(q,q)∃qP(q,q) to be inferred from the premise ∀x∃yP(x,y)∀x∃yP(x,y).  
4. A common error among students is to suppose that, in unification, one is allowed to substitute a term for a Skolem constant instead of for a variable. For instance, they will say that the formulas P(Sk1)P(Sk1) and P(A)P(A) can be unified under the substitution {Sk1/A}{Sk1/A}. Give an example where this leads to an invalid inference.

[Exercise 9](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_9/)

This question considers Horn KBs, such as the following:

P(F(x))⇒P(x)Q(x)⇒P(F(x))P(A)Q(B)P(F(x))⇒P(x)Q(x)⇒P(F(x))P(A)Q(B)

Let FC be a breadth-first forward-chaining algorithm that repeatedly adds all consequences of currently satisfied rules; let BC be a depth-first left-to-right backward-chaining algorithm that tries clauses in the order given in the KB. Which of the following are true?  
1. FC will infer the literal Q(A)Q(A).  
2. FC will infer the literal P(B)P(B).  
3. If FC has failed to infer a given literal, then it is not entailed by the KB.  
4. BC will return truetrue given the query P(B)P(B).  
5. If BC does not return truetrue given a query literal, then it is not entailed by the KB.

[Exercise 10 (csp-clause-exercise)](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_10/)

Explain how to write any given 3-SAT problem of arbitrary size using a single first-order definite clause and no more than 30 ground facts.

[Exercise 11](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_11/)

Suppose you are given the following axioms:  
1. 0≤30≤3.  
2. 7≤97≤9.  
3. ∀xx≤x∀xx≤x.  
4. ∀xx≤x+0∀xx≤x+0.  
5. ∀xx+0≤x∀xx+0≤x.  
6. ∀x,yx+y≤y+x∀x,yx+y≤y+x.  
7. ∀w,x,y,zw≤y∀w,x,y,zw≤y ∧∧ x≤zx≤z ⇒⇒ w+x≤y+zw+x≤y+z.  
8. ∀x,y,zx≤y∧y≤z⇒x≤z∀x,y,zx≤y∧y≤z⇒x≤z  
  
1. Give a backward-chaining proof of the sentence 7≤3+97≤3+9. (Be sure, of course, to use only the axioms given here, not anything else you may know about arithmetic.) Show only the steps that leads to success, not the irrelevant steps.  
2. Give a forward-chaining proof of the sentence 7≤3+97≤3+9. Again, show only the steps that lead to success.

[Exercise 12](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_12/)

Suppose you are given the following axioms:  
> 1. 0≤40≤4.  
> 2. 5≤95≤9.  
> 3. ∀xx≤x∀xx≤x.  
> 4. ∀xx≤x+0∀xx≤x+0.  
> 5. ∀xx+0≤x∀xx+0≤x.  
> 6. ∀x,yx+y≤y+x∀x,yx+y≤y+x.  
> 7. ∀w,x,y,zw≤y∀w,x,y,zw≤y ∧∧ x≤z⇒x≤z⇒ w+x≤y+zw+x≤y+z.  
> 8. ∀x,y,zx≤y∧y≤z⇒x≤z∀x,y,zx≤y∧y≤z⇒x≤z  
  
1. Give a backward-chaining proof of the sentence 5≤4+95≤4+9. (Be sure, of course, to use only the axioms given here, not anything else you may know about arithmetic.) Show only the steps that leads to success, not the irrelevant steps.  
2. Give a forward-chaining proof of the sentence 5≤4+95≤4+9. Again, show only the steps that lead to success.

[Exercise 13](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_13/)

A popular children’s riddle is “Brothers and sisters have I none, but that man’s father is my father’s son.” Use the rules of the family domain (Section  on page  to show who that man is. You may apply any of the inference methods described in this chapter. Why do you think that this riddle is difficult?

[Exercise 14](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_14/)

Suppose we put into a logical knowledge base a segment of the U.S. census data listing the age, city of residence, date of birth, and mother of every person, using social security numbers as identifying constants for each person. Thus, George’s age is given by Age(443−65−1282,56)Age(443−65−1282,56). Which of the following indexing schemes S1–S5 enable an efficient solution for which of the queries Q1–Q4 (assuming normal backward chaining)?  
  
- **S1**: an index for each atom in each position.  
- **S2**: an index for each first argument.  
- **S3**: an index for each predicate atom.  
- **S4**: an index for each *combination* of predicate and first argument.  
- **S5**: an index for each *combination* of predicate and second argument and an index for each first argument.  
- **Q1**: Age(443−44−4321,x)Age(443−44−4321,x)  
- **Q2**: ResidesIn(x,Houston)ResidesIn(x,Houston)  
- **Q3**: Mother(x,y)Mother(x,y)  
- **Q4**: Age(x,34)∧ResidesIn(x,TinyTownUSA)Age(x,34)∧ResidesIn(x,TinyTownUSA)

[Exercise 15 (standardize-failure-exercise)](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_15/)

One might suppose that we can avoid the problem of variable conflict in unification during backward chaining by standardizing apart all of the sentences in the knowledge base once and for all. Show that, for some sentences, this approach cannot work. (*Hint*: Consider a sentence in which one part unifies with another.)

[Exercise 16](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_16/)

In this exercise, use the sentences you wrote in Exercise [fol-horses-exercise](https://aimacode.github.io/aima-exercises/logical-inference-exercises/) to answer a question by using a backward-chaining algorithm.  
1. Draw the proof tree generated by an exhaustive backward-chaining algorithm for the query ∃hHorse(h)∃hHorse(h), where clauses are matched in the order given.  
2. What do you notice about this domain?  
3. How many solutions for hh actually follow from your sentences?  
4. Can you think of a way to find all of them? (*Hint*: See [Smith+al:1986](https://aimacode.github.io/aima-exercises/logical-inference-exercises/).)

[Exercise 17 (bc-trace-exercise)](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_17/)

Trace the execution of the backward-chaining algorithm in Figure [9.6](https://aimacode.github.io/aima-exercises/figures/backward-chaining-algorithm) (page  when it is applied to solve the crime problem (page [west-problem-page](https://aimacode.github.io/aima-exercises/logical-inference-exercises/). Show the sequence of values taken on by the goalsgoals variable, and arrange them into a tree.

[Exercise 18](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_18/)

The following Prolog code defines a predicate P. (Remember that uppercase terms are variables, not constants, in Prolog.)  
P(X,[X|Y]).  
P(X,[Y|Z]) :- P(X,Z).  
1. Show proof trees and solutions for the queries P(A,[2,1,3]) and P(2,[1,A,3]).  
2. What standard list operation does P represent?

[Exercise 19](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_19/)

The following Prolog code defines a predicate P. (Remember that uppercase terms are variables, not constants, in Prolog.)  
P(X,[X|Y]).  
P(X,[Y|Z]) :- P(X,Z).  
1. Show proof trees and solutions for the queries P(A,[1,2,3]) and P(2,[1,A,3]).  
2. What standard list operation does P represent?

[Exercise 20](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_20/)

This exercise looks at sorting in Prolog.  
1. Write Prolog clauses that define the predicate sorted(L), which is true if and only if list L is sorted in ascending order.  
2. Write a Prolog definition for the predicate perm(L,M), which is true if and only if L is a permutation of M.  
3. Define sort(L,M) (M is a sorted version of L) using perm and sorted.  
4. Run sort on longer and longer lists until you lose patience. What is the time complexity of your program?  
5. Write a faster sorting algorithm, such as insertion sort or quicksort, in Prolog.

[Exercise 21 (diff-simplify-exercise)](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_21/)

This exercise looks at the recursive application of rewrite rules, using logic programming. A rewrite rule (or **demodulator** in terminology) is an equation with a specified direction. For example, the rewrite rule x+0→xx+0→x suggests replacing any expression that matches x+0x+0 with the expression xx. Rewrite rules are a key component of equational reasoning systems. Use the predicate rewrite(X,Y) to represent rewrite rules. For example, the earlier rewrite rule is written as rewrite(X+0,X). Some terms are *primitive* and cannot be further simplified; thus, we write primitive(0) to say that 0 is a primitive term.  
1. Write a definition of a predicate simplify(X,Y), that is true when Y is a simplified version of X—that is, when no further rewrite rules apply to any subexpression of Y.  
2. Write a collection of rules for the simplification of expressions involving arithmetic operators, and apply your simplification algorithm to some sample expressions.  
3. Write a collection of rewrite rules for symbolic differentiation, and use them along with your simplification rules to differentiate and simplify expressions involving arithmetic expressions, including exponentiation.

[Exercise 22](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_22/)

This exercise considers the implementation of search algorithms in Prolog. Suppose that successor(X,Y) is true when state Y is a successor of state X; and that goal(X) is true when X is a goal state. Write a definition for solve(X,P), which means that P is a path (list of states) beginning with X, ending in a goal state, and consisting of a sequence of legal steps as defined by successor. You will find that depth-first search is the easiest way to do this. How easy would it be to add heuristic search control?

[Exercise 23](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_23/)

Suppose a knowledge base contains just the following first-order Horn clauses:

Ancestor(Mother(x),x)Ancestor(Mother(x),x)

Ancestor(x,y)∧Ancestor(y,z)⟹Ancestor(x,z)Ancestor(x,y)∧Ancestor(y,z)⟹Ancestor(x,z)

Consider a forward chaining algorithm that, on the jjth iteration, terminates if the KB contains a sentence that unifies with the query, else adds to the KB every atomic sentence that can be inferred from the sentences already in the KB after iteration j−1j−1.  
1. For each of the following queries, say whether the algorithm will (1) give an answer (if so, write down that answer); or (2) terminate with no answer; or (3) never terminate.  
1. Ancestor(Mother(y),John)Ancestor(Mother(y),John)  
2. Ancestor(Mother(Mother(y)),John)Ancestor(Mother(Mother(y)),John)  
3. Ancestor(Mother(Mother(Mother(y))),Mother(y))Ancestor(Mother(Mother(Mother(y))),Mother(y))  
4. Ancestor(Mother(John),Mother(Mother(John)))Ancestor(Mother(John),Mother(Mother(John)))  
2. Can a resolution algorithm prove the sentence ¬Ancestor(John,John)¬Ancestor(John,John) from the original knowledge base? Explain how, or why not.  
3. Suppose we add the assertion that ¬(Mother(x)x)¬(Mother(x)x) and augment the resolution algorithm with inference rules for equality. Now what is the answer to (b)?

[Exercise 24](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_24/)

Let LL be the first-order language with a single predicate S(p,q)S(p,q), meaning “pp shaves  qq.” Assume a domain of people.  
1. Consider the sentence “There exists a person PP who shaves every one who does not shave themselves, and only people that do not shave themselves.” Express this in LL.  
2. Convert the sentence in (a) to clausal form.  
3. Construct a resolution proof to show that the clauses in (b) are inherently inconsistent. (Note: you do not need any additional axioms.)

[Exercise 25](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_25/)

How can resolution be used to show that a sentence is valid? Unsatisfiable?

[Exercise 26](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_26/)

Construct an example of two clauses that can be resolved together in two different ways giving two different outcomes.

[Exercise 27](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_27/)

From “Horses are animals,” it follows that “The head of a horse is the head of an animal.” Demonstrate that this inference is valid by carrying out the following steps:  
1. Translate the premise and the conclusion into the language of first-order logic. Use three predicates: HeadOf(h,x)HeadOf(h,x) (meaning “hh is the head of xx”), Horse(x)Horse(x), and Animal(x)Animal(x).  
2. Negate the conclusion, and convert the premise and the negated conclusion into conjunctive normal form.  
3. Use resolution to show that the conclusion follows from the premise.

[Exercise 28](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_28/)

From “Sheep are animals,” it follows that “The head of a sheep is the head of an animal.” Demonstrate that this inference is valid by carrying out the following steps:  
1. Translate the premise and the conclusion into the language of first-order logic. Use three predicates: HeadOf(h,x)HeadOf(h,x) (meaning “hh is the head of xx”), Sheep(x)Sheep(x), and Animal(x)Animal(x).  
2. Negate the conclusion, and convert the premise and the negated conclusion into conjunctive normal form.  
3. Use resolution to show that the conclusion follows from the premise.

[Exercise 29 (quantifier-order-exercise)](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_29/)

Here are two sentences in the language of first-order logic:  
- **(A)** ∀x∃y(x≥y)∀x∃y(x≥y) - **(B)** ∃y∀x(x≥y)∃y∀x(x≥y) 1. Assume that the variables range over all the natural numbers 0,1,2,…,∞0,1,2,…,∞ and that the “≥≥” predicate means “is greater than or equal to.” Under this interpretation, translate (A) and (B) into English.  
2. Is (A) true under this interpretation?  
3. Is (B) true under this interpretation?  
4. Does (A) logically entail (B)?  
5. Does (B) logically entail (A)?  
6. Using resolution, try to prove that (A) follows from (B). Do this even if you think that (B) does not logically entail (A); continue until the proof breaks down and you cannot proceed (if it does break down). Show the unifying substitution for each resolution step. If the proof fails, explain exactly where, how, and why it breaks down.  
7. Now try to prove that (B) follows from (A).

[Exercise 30](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_30/)

Resolution can produce nonconstructive proofs for queries with variables, so we had to introduce special mechanisms to extract definite answers. Explain why this issue does not arise with knowledge bases containing only definite clauses.

[Exercise 31](https://aimacode.github.io/aima-exercises/logical-inference-exercises/ex_31/)

We said in this chapter that resolution cannot be used to generate all logical consequences of a set of sentences. Can any algorithm do this?

10. Classical Planning

[Exercise 1](https://aimacode.github.io/aima-exercises/planning-exercises/ex_1/)

Consider a robot whose operation is described by the following PDDL operators:

Op(Go(x,y),At(Robot,x),¬At(Robot,x)∧At(Robot,y))Op(Go(x,y),At(Robot,x),¬At(Robot,x)∧At(Robot,y))

Op(Pick(o),At(Robot,x)∧At(o,x),¬At(o,x)∧Holding(o))Op(Pick(o),At(Robot,x)∧At(o,x),¬At(o,x)∧Holding(o))

Op(Drop(o),At(Robot,x)∧Holding(o),At(o,x)∧¬Holding(o)Op(Drop(o),At(Robot,x)∧Holding(o),At(o,x)∧¬Holding(o)

1. The operators allow the robot to hold more than one object. Show how to modify them with an EmptyHandEmptyHand predicate for a robot that can hold only one object.  
2. Assuming that these are the only actions in the world, write a successor-state axiom for EmptyHandEmptyHand.

[Exercise 2](https://aimacode.github.io/aima-exercises/planning-exercises/ex_2/)

Describe the differences and similarities between problem solving and planning.

[Exercise 3 (strips-airport-exercise)](https://aimacode.github.io/aima-exercises/planning-exercises/ex_3/)

Given the action schemas and initial state from Figure [10.1](https://aimacode.github.io/aima-exercises/figures/airport-pddl-algorithm.png), what are all the applicable concrete instances of Fly(p,from,to)Fly(p,from,to) in the state described by

At(P1,JFK)∧At(P2,SFO)∧Plane(P1)∧Plane(P2)∧Airport(JFK)∧Airport(SFO)?At(P1,JFK)∧At(P2,SFO)∧Plane(P1)∧Plane(P2)∧Airport(JFK)∧Airport(SFO)?

[Exercise 4](https://aimacode.github.io/aima-exercises/planning-exercises/ex_4/)

The monkey-and-bananas problem is faced by a monkey in a laboratory with some bananas hanging out of reach from the ceiling. A box is available that will enable the monkey to reach the bananas if he climbs on it. Initially, the monkey is at AA, the bananas at BB, and the box at CC. The monkey and box have height LowLow, but if the monkey climbs onto the box he will have height HighHigh, the same as the bananas. The actions available to the monkey include GoGo from one place to another, PushPush an object from one place to another, ClimbUpClimbUp onto or ClimbDownClimbDown from an object, and GraspGrasp or UngraspUngrasp an object. The result of a GraspGrasp is that the monkey holds the object if the monkey and object are in the same place at the same height.  
1. Write down the initial state description.  
2. Write the six action schemas.  
3. Suppose the monkey wants to fool the scientists, who are off to tea, by grabbing the bananas, but leaving the box in its original place. Write this as a general goal (i.e., not assuming that the box is necessarily at C) in the language of situation calculus. Can this goal be solved by a classical planning system?  
4. Your schema for pushing is probably incorrect, because if the object is too heavy, its position will remain the same when the PushPush schema is applied. Fix your action schema to account for heavy objects.

[Exercise 5](https://aimacode.github.io/aima-exercises/planning-exercises/ex_5/)

The original {Strips} planner was designed to control Shakey the robot. Figure [shakey-figure](https://aimacode.github.io/aima-exercises/planning-exercises/" \l "shakey-figure) shows a version of Shakey’s world consisting of four rooms lined up along a corridor, where each room has a door and a light switch. The actions in Shakey’s world include moving from place to place, pushing movable objects (such as boxes), climbing onto and down from rigid objects (such as boxes), and turning light switches on and off. The robot itself could not climb on a box or toggle a switch, but the planner was capable of finding and printing out plans that were beyond the robot’s abilities. Shakey’s six actions are the following:  
- Go(x,y,r)Go(x,y,r), which requires that Shakey be AtAt xx and that xx and yy are locations InIn the same room rr. By convention a door between two rooms is in both of them.  
- Push a box bb from location xx to location yy within the same room: Push(b,x,y,r)Push(b,x,y,r). You will need the predicate BoxBox and constants for the boxes.  
- Climb onto a box from position xx: ClimbUp(x,b)ClimbUp(x,b); climb down from a box to position xx: ClimbDown(b,x)ClimbDown(b,x). We will need the predicate OnOn and the constant FloorFloor.  
- Turn a light switch on or off: TurnOn(s,b)TurnOn(s,b); TurnOff(s,b)TurnOff(s,b). To turn a light on or off, Shakey must be on top of a box at the light switch’s location.  
Write PDDL sentences for Shakey’s six actions and the initial state from Construct a plan for Shakey to get Box2Box2 into Room2Room2.

**Shakey's world. Shakey can move between landmarks within a room, can pass through the door between rooms, can climb climbable objects and push pushable objects, and can flip light switches.**

[Exercise 6](https://aimacode.github.io/aima-exercises/planning-exercises/ex_6/)

A finite Turing machine has a finite one-dimensional tape of cells, each cell containing one of a finite number of symbols. One cell has a read and write head above it. There is a finite set of states the machine can be in, one of which is the accept state. At each time step, depending on the symbol on the cell under the head and the machine’s current state, there are a set of actions we can choose from. Each action involves writing a symbol to the cell under the head, transitioning the machine to a state, and optionally moving the head left or right. The mapping that determines which actions are allowed is the Turing machine’s program. Your goal is to control the machine into the accept state.  
  
Represent the Turing machine acceptance problem as a planning problem. If you can do this, it demonstrates that determining whether a planning problem has a solution is at least as hard as the Turing acceptance problem, which is PSPACE-hard.

[Exercise 7 (negative-effects-exercise)](https://aimacode.github.io/aima-exercises/planning-exercises/ex_7/)

Explain why dropping negative effects from every action schema results in a relaxed problem, provided that preconditions and goals contain only positive literals.

[Exercise 8 (sussman-anomaly-exercise)](https://aimacode.github.io/aima-exercises/planning-exercises/ex_8/)

Figure [10.4](https://aimacode.github.io/aima-exercises/figures/sussman-anamoly-figure.png) (page ) shows a blocks-world problem that is known as the {Sussman anomaly}. The problem was considered anomalous because the noninterleaved planners of the early 1970s could not solve it. Write a definition of the problem and solve it, either by hand or with a planning program. A noninterleaved planner is a planner that, when given two subgoals G1G1 and G2G2, produces either a plan for G1G1 concatenated with a plan for G2G2, or vice versa. Can a noninterleaved planner solve this problem? How, or why not?

[Exercise 9](https://aimacode.github.io/aima-exercises/planning-exercises/ex_9/)

Prove that backward search with PDDL problems is complete.

[Exercise 10](https://aimacode.github.io/aima-exercises/planning-exercises/ex_10/)

Construct levels 0, 1, and 2 of the planning graph for the problem in Figure [10.1](https://aimacode.github.io/aima-exercises/figures/airport-pddl-algorithm.png)

[Exercise 11 (graphplan-proof-exercise)](https://aimacode.github.io/aima-exercises/planning-exercises/ex_11/)

Prove the following assertions about planning graphs:  
1. A literal that does not appear in the final level of the graph cannot be achieved.  
2. The level cost of a literal in a serial graph is no greater than the actual cost of an optimal plan for achieving it.

[Exercise 12](https://aimacode.github.io/aima-exercises/planning-exercises/ex_12/)

We saw that planning graphs can handle only propositional actions. What if we want to use planning graphs for a problem with variables in the goal, such as At(P1,x)∧At(P2,x)At(P1,x)∧At(P2,x), where xx is assumed to be bound by an existential quantifier that ranges over a finite domain of locations? How could you encode such a problem to work with planning graphs?

[Exercise 13](https://aimacode.github.io/aima-exercises/planning-exercises/ex_13/)

The set-level heuristic (see page  uses a planning graph to estimate the cost of achieving a conjunctive goal from the current state. What relaxed problem is the set-level heuristic the solution to?

[Exercise 14](https://aimacode.github.io/aima-exercises/planning-exercises/ex_14/)

Examine the definition of **bidirectional search** in Chapter [Solving Problems by Searching](https://aimacode.github.io/aima-exercises/search-exercises/).  
1. Would bidirectional state-space search be a good idea for planning?  
2. What about bidirectional search in the space of partial-order plans?  
3. Devise a version of partial-order planning in which an action can be added to a plan if its preconditions can be achieved by the effects of actions already in the plan. Explain how to deal with conflicts and ordering constraints. Is the algorithm essentially identical to forward state-space search?

[Exercise 15](https://aimacode.github.io/aima-exercises/planning-exercises/ex_15/)

We contrasted forward and backward state-space searchers with partial-order planners, saying that the latter is a plan-space searcher. Explain how forward and backward state-space search can also be considered plan-space searchers, and say what the plan refinement operators are.

[Exercise 16 (satplan-preconditions-exercise)](https://aimacode.github.io/aima-exercises/planning-exercises/ex_16/)

Up to now we have assumed that the plans we create always make sure that an action’s preconditions are satisfied. Let us now investigate what propositional successor-state axioms such as HaveArrowt+1⇔HaveArrowt+1⇔ (HaveArrowt∧¬Shoott)(HaveArrowt∧¬Shoott) have to say about actions whose preconditions are not satisfied.  
1. Show that the axioms predict that nothing will happen when an action is executed in a state where its preconditions are not satisfied.  
2. Consider a plan pp that contains the actions required to achieve a goal but also includes illegal actions. Is it the case that

initialstate∧successor−stateaxioms∧p⊨goal?initialstate∧successor−stateaxioms∧p⊨goal?

3. With first-order successor-state axioms in situation calculus, is it possible to prove that a plan containing illegal actions will achieve the goal?

[Exercise 17 (strips-translation-exercise)](https://aimacode.github.io/aima-exercises/planning-exercises/ex_17/)

Consider how to translate a set of action schemas into the successor-state axioms of situation calculus.  
1. Consider the schema for Fly(p,from,to)Fly(p,from,to). Write a logical definition for the predicate Poss(Fly(p,from,to),s)Poss(Fly(p,from,to),s), which is true if the preconditions for Fly(p,from,to)Fly(p,from,to) are satisfied in situation ss.  
2. Next, assuming that Fly(p,from,to)Fly(p,from,to) is the only action schema available to the agent, write down a successor-state axiom for At(p,x,s)At(p,x,s) that captures the same information as the action schema.  
3. Now suppose there is an additional method of travel: Teleport(p,from,to)Teleport(p,from,to). It has the additional precondition ¬Warped(p)¬Warped(p) and the additional effect Warped(p)Warped(p). Explain how the situation calculus knowledge base must be modified.  
4. Finally, develop a general and precisely specified procedure for carrying out the translation from a set of action schemas to a set of successor-state axioms.

[Exercise 18 (disjunctive-satplan-exercise)](https://aimacode.github.io/aima-exercises/planning-exercises/ex_18/)

In the SATPlanSATPlan algorithm in Figure [7.22](https://aimacode.github.io/aima-exercises/figures/satplan-agent-algorithm.png) (page , each call to the satisfiability algorithm asserts a goal gTgT, where TT ranges from 0 to TmaxTmax. Suppose instead that the satisfiability algorithm is called only once, with the goal g0∨g1∨⋯∨gTmaxg0∨g1∨⋯∨gTmax.  
1. Will this always return a plan if one exists with length less than or equal to TmaxTmax?  
2. Does this approach introduce any new spurious “solutions”?  
3. Discuss how one might modify a satisfiability algorithm such as WalkSATWalkSAT so that it finds short solutions (if they exist) when given a disjunctive goal of this form.

11. Planning and Acting in the Real World

[Exercise 1](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_1/)

The goals we have considered so far all ask the planner to make the world satisfy the goal at just one time step. Not all goals can be expressed this way: you do not achieve the goal of suspending a chandelier above the ground by throwing it in the air. More seriously, you wouldn’t want your spacecraft life-support system to supply oxygen one day but not the next. A *maintenance goal* is achieved when the agent’s plan causes a condition to hold continuously from a given state onward. Describe how to extend the formalism of this chapter to support maintenance goals.

[Exercise 2](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_2/)

You have a number of trucks with which to deliver a set of packages. Each package starts at some location on a grid map, and has a destination somewhere else. Each truck is directly controlled by moving forward and turning. Construct a hierarchy of high-level actions for this problem. What knowledge about the solution does your hierarchy encode?

[Exercise 3 (HLA-unique-exercise)](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_3/)

Suppose that a high-level action has exactly one implementation as a sequence of primitive actions. Give an algorithm for computing its preconditions and effects, given the complete refinement hierarchy and schemas for the primitive actions.

[Exercise 4](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_4/)

Suppose that the optimistic reachable set of a high-level plan is a superset of the goal set; can anything be concluded about whether the plan achieves the goal? What if the pessimistic reachable set doesn’t intersect the goal set? Explain.

[Exercise 5 (HLA-progression-exercise)](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_5/)

Write an algorithm that takes an initial state (specified by a set of propositional literals) and a sequence of HLAs (each defined by preconditions and angelic specifications of optimistic and pessimistic reachable sets) and computes optimistic and pessimistic descriptions of the reachable set of the sequence.

[Exercise 6](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_6/)

In Figure [11.2](https://aimacode.github.io/aima-exercises/figures/jobshop-cpm-figure.png) we showed how to describe actions in a scheduling problem by using separate fields for , , and . Now suppose we wanted to combine scheduling with nondeterministic planning, which requires nondeterministic and conditional effects. Consider each of the three fields and explain if they should remain separate fields, or if they should become effects of the action. Give an example for each of the three.

[Exercise 7](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_7/)

Some of the operations in standard programming languages can be modeled as actions that change the state of the world. For example, the assignment operation changes the contents of a memory location, and the print operation changes the state of the output stream. A program consisting of these operations can also be considered as a plan, whose goal is given by the specification of the program. Therefore, planning algorithms can be used to construct programs that achieve a given specification.  
1. Write an action schema for the assignment operator (assigning the value of one variable to another). Remember that the original value will be overwritten!  
2. Show how object creation can be used by a planner to produce a plan for exchanging the values of two variables by using a temporary variable.

[Exercise 8](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_8/)

Consider the following argument: In a framework that allows uncertain initial states, **nondeterministic effects** are just a notational convenience, not a source of additional representational power. For any action schema aa with nondeterministic effect P∨QP∨Q, we could always replace it with the conditional effects  R: P∧ ¬R: Q R: P∧ ¬R: Q, which in turn can be reduced to two regular actions. The proposition RR stands for a random proposition that is unknown in the initial state and for which there are no sensing actions. Is this argument correct? Consider separately two cases, one in which only one instance of action schema aa is in the plan, the other in which more than one instance is.

[Exercise 9 (conformant-flip-literal-exercise)](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_9/)

Suppose the FlipFlip action always changes the truth value of variable LL. Show how to define its effects by using an action schema with conditional effects. Show that, despite the use of conditional effects, a 1-CNF belief state representation remains in 1-CNF after a FlipFlip.

[Exercise 10](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_10/)

In the blocks world we were forced to introduce two action schemas, MoveMove and MoveToTableMoveToTable, in order to maintain the ClearClear predicate properly. Show how conditional effects can be used to represent both of these cases with a single action.

[Exercise 11 (alt-vacuum-exercise)](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_11/)

Conditional effects were illustrated for the SuckSuck action in the vacuum world—which square becomes clean depends on which square the robot is in. Can you think of a new set of propositional variables to define states of the vacuum world, such that SuckSuck has an *unconditional* description? Write out the descriptions of SuckSuck, LeftLeft, and RightRight, using your propositions, and demonstrate that they suffice to describe all possible states of the world.

[Exercise 12](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_12/)

Find a suitably dirty carpet, free of obstacles, and vacuum it. Draw the path taken by the vacuum cleaner as accurately as you can. Explain it, with reference to the forms of planning discussed in this chapter.

[Exercise 13](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_13/)

The following quotes are from the backs of shampoo bottles. Identify each as an unconditional, conditional, or execution-monitoring plan. (a) “Lather. Rinse. Repeat.” (b) “Apply shampoo to scalp and let it remain for several minutes. Rinse and repeat if necessary.” (c) “See a doctor if problems persist.”

[Exercise 14](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_14/)

Consider the following problem: A patient arrives at the doctor’s office with symptoms that could have been caused either by dehydration or by disease DD (but not both). There are two possible actions: DrinkDrink, which unconditionally cures dehydration, and MedicateMedicate, which cures disease DD but has an undesirable side effect if taken when the patient is dehydrated. Write the problem description, and diagram a sensorless plan that solves the problem, enumerating all relevant possible worlds.

[Exercise 15](https://aimacode.github.io/aima-exercises/advanced-planning-exercises/ex_15/)

To the medication problem in the previous exercise, add a TestTest action that has the conditional effect CultureGrowthCultureGrowth when DiseaseDisease is true and in any case has the perceptual effect Known(CultureGrowth)Known(CultureGrowth). Diagram a conditional plan that solves the problem and minimizes the use of the MedicateMedicate action.

12. Knowledge Representation

[Exercise 1](https://aimacode.github.io/aima-exercises/kr-exercises/ex_1/)

Define an ontology in first-order logic for tic-tac-toe. The ontology should contain situations, actions, squares, players, marks (X, O, or blank), and the notion of winning, losing, or drawing a game. Also define the notion of a forced win (or draw): a position from which a player can force a win (or draw) with the right sequence of actions. Write axioms for the domain. (Note: The axioms that enumerate the different squares and that characterize the winning positions are rather long. You need not write these out in full, but indicate clearly what they look like.)

[Exercise 2](https://aimacode.github.io/aima-exercises/kr-exercises/ex_2/)

You are to create a system for advising computer science undergraduates on what courses to take over an extended period in order to satisfy the program requirements. (Use whatever requirements are appropriate for your institution.) First, decide on a vocabulary for representing all the information, and then represent it; then formulate a query to the system that will return a legal program of study as a solution. You should allow for some tailoring to individual students, in that your system should ask what courses or equivalents the student has already taken, and not generate programs that repeat those courses. Suggest ways in which your system could be improved—for example to take into account knowledge about student preferences, the workload, good and bad instructors, and so on. For each kind of knowledge, explain how it could be expressed logically. Could your system easily incorporate this information to find all feasible programs of study for a student? Could it find the *best* program?

[Exercise 3](https://aimacode.github.io/aima-exercises/kr-exercises/ex_3/)

Figure [12.1](https://aimacode.github.io/aima-exercises/figures/ontology-figure.png) shows the top levels of a hierarchy for everything. Extend it to include as many real categories as possible. A good way to do this is to cover all the things in your everyday life. This includes objects and events. Start with waking up, and proceed in an orderly fashion noting everything that you see, touch, do, and think about. For example, a random sampling produces music, news, milk, walking, driving, gas, Soda Hall, carpet, talking, Professor Fateman, chicken curry, tongue, $ 7, sun, the daily newspaper, and so on.  
You should produce both a single hierarchy chart (on a large sheet of paper) and a listing of objects and categories with the relations satisfied by members of each category. Every object should be in a category, and every category should be in the hierarchy.

[Exercise 4 (windows-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_4/)

Develop a representational system for reasoning about windows in a window-based computer interface. In particular, your representation should be able to describe:  
- The state of a window: minimized, displayed, or nonexistent.  
- Which window (if any) is the active window.  
- The position of every window at a given time.  
- The order (front to back) of overlapping windows.  
- The actions of creating, destroying, resizing, and moving windows; changing the state of a window; and bringing a window to the front. Treat these actions as atomic; that is, do not deal with the issue of relating them to mouse actions. Give axioms describing the effects of actions on fluents. You may use either event or situation calculus.  
Assume an ontology containing *situations,* *actions,* *integers* (for xx and yy coordinates) and *windows*. Define a language over this ontology; that is, a list of constants, function symbols, and predicates with an English description of each. If you need to add more categories to the ontology (e.g., pixels), you may do so, but be sure to specify these in your write-up. You may (and should) use symbols defined in the text, but be sure to list these explicitly.

[Exercise 5](https://aimacode.github.io/aima-exercises/kr-exercises/ex_5/)

State the following in the language you developed for the previous exercise:  
1. In situation S0S0, window W1W1 is behind W2W2 but sticks out on the top and bottom. Do *not* state exact coordinates for these; describe the *general* situation.  
2. If a window is displayed, then its top edge is higher than its bottom edge.  
3. After you create a window ww, it is displayed.  
4. A window can be minimized only if it is displayed.

[Exercise 6](https://aimacode.github.io/aima-exercises/kr-exercises/ex_6/)

State the following in the language you developed for the previous exercise:  
1. In situation S0S0, window W1W1 is behind W2W2 but sticks out on the top and bottom. Do *not* state exact coordinates for these; describe the *general* situation.  
2. If a window is displayed, then its top edge is higher than its bottom edge.  
3. After you create a window ww, it is displayed.  
4. A window can be minimized only if it is displayed.

[Exercise 7](https://aimacode.github.io/aima-exercises/kr-exercises/ex_7/)

(Adapted from an example by Doug Lenat.) Your mission is to capture, in logical form, enough knowledge to answer a series of questions about the following simple scenario:  
  
*Yesterday John went to the North Berkeley Safeway supermarket and*  
*bought two pounds of tomatoes and a pound of ground beef.* Start by trying to represent the content of the sentence as a series of assertions. You should write sentences that have straightforward logical structure (e.g., statements that objects have certain properties, that objects are related in certain ways, that all objects satisfying one property satisfy another). The following might help you get started:  
- Which classes, objects, and relations would you need? What are their parents, siblings and so on? (You will need events and temporal ordering, among other things.)  
- Where would they fit in a more general hierarchy?  
- What are the constraints and interrelationships among them?  
- How detailed must you be about each of the various concepts?  
To answer the questions below, your knowledge base must include background knowledge. You’ll have to deal with what kind of things are at a supermarket, what is involved with purchasing the things one selects, what the purchases will be used for, and so on. Try to make your representation as general as possible. To give a trivial example: don’t say “People buy food from Safeway,” because that won’t help you with those who shop at another supermarket. Also, don’t turn the questions into answers; for example, question (c) asks “Did John buy any meat?”—not “Did John buy a pound of ground beef?”  
Sketch the chains of reasoning that would answer the questions. If possible, use a logical reasoning system to demonstrate the sufficiency of your knowledge base. Many of the things you write might be only approximately correct in reality, but don’t worry too much; the idea is to extract the common sense that lets you answer these questions at all. A truly complete answer to this question is *extremely* difficult, probably beyond the state of the art of current knowledge representation. But you should be able to put together a consistent set of axioms for the limited questions posed here.  
1. Is John a child or an adult? [Adult]  
2. Does John now have at least two tomatoes? [Yes]  
3. Did John buy any meat? [Yes]  
4. If Mary was buying tomatoes at the same time as John, did he see her? [Yes]  
5. Are the tomatoes made in the supermarket? [No]  
6. What is John going to do with the tomatoes? [Eat them]  
7. Does Safeway sell deodorant? [Yes]  
8. Did John bring some money or a credit card to the supermarket? [Yes]  
9. Does John have less money after going to the supermarket? [Yes]

[Exercise 8](https://aimacode.github.io/aima-exercises/kr-exercises/ex_8/)

Make the necessary additions or changes to your knowledge base from the previous exercise so that the questions that follow can be answered. Include in your report a discussion of your changes, explaining why they were needed, whether they were minor or major, and what kinds of questions would necessitate further changes.  
1. Are there other people in Safeway while John is there? [Yes—staff!]  
2. Is John a vegetarian? [No]  
3. Who owns the deodorant in Safeway? [Safeway Corporation]  
4. Did John have an ounce of ground beef? [Yes]  
5. Does the Shell station next door have any gas? [Yes]  
6. Do the tomatoes fit in John’s car trunk? [Yes]

[Exercise 9](https://aimacode.github.io/aima-exercises/kr-exercises/ex_9/)

Represent the following seven sentences using and extending the representations developed in the chapter:  
1. Water is a liquid between 0 and 100 degrees.  
2. Water boils at 100 degrees.  
3. The water in John’s water bottle is frozen.  
4. Perrier is a kind of water.  
5. John has Perrier in his water bottle.  
6. All liquids have a freezing point.  
7. A liter of water weighs more than a liter of alcohol.

[Exercise 10 (part-decomposition-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_10/)

Write definitions for the following:  
1. ExhaustivePartDecompositionExhaustivePartDecomposition  
2. PartPartitionPartPartition  
3. PartwiseDisjointPartwiseDisjoint  
These should be analogous to the definitions for ExhaustiveDecompositionExhaustiveDecomposition, PartitionPartition, and DisjointDisjoint. Is it the case that PartPartition(s,BunchOf(s))PartPartition(s,BunchOf(s))? If so, prove it; if not, give a counterexample and define sufficient conditions under which it does hold.

[Exercise 11 (alt-measure-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_11/)

An alternative scheme for representing measures involves applying the units function to an abstract length object. In such a scheme, one would write Inches(Length(L1))=1.5Inches(Length(L1))=1.5. How does this scheme compare with the one in the chapter? Issues include conversion axioms, names for abstract quantities (such as “50 dollars”), and comparisons of abstract measures in different units (50 inches is more than 50 centimeters).

[Exercise 12](https://aimacode.github.io/aima-exercises/kr-exercises/ex_12/)

Write a set of sentences that allows one to calculate the price of an individual tomato (or other object), given the price per pound. Extend the theory to allow the price of a bag of tomatoes to be calculated.

[Exercise 13 (namematch-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_13/)

Add sentences to extend the definition of the predicate Name(s,c)Name(s,c) so that a string such as “laptop computer” matches the appropriate category names from a variety of stores. Try to make your definition general. Test it by looking at ten online stores, and at the category names they give for three different categories. For example, for the category of laptops, we found the names “Notebooks,” “Laptops,” “Notebook Computers,” “Notebook,” “Laptops and Notebooks,” and “Notebook PCs.” Some of these can be covered by explicit NameName facts, while others could be covered by sentences for handling plurals, conjunctions, etc.

[Exercise 14](https://aimacode.github.io/aima-exercises/kr-exercises/ex_14/)

Write event calculus axioms to describe the actions in the wumpus world.

[Exercise 15](https://aimacode.github.io/aima-exercises/kr-exercises/ex_15/)

State the interval-algebra relation that holds between every pair of the following real-world events:  
> LKLK: The life of President Kennedy.  
> IKIK: The infancy of President Kennedy.  
> PKPK: The presidency of President Kennedy.  
> LJLJ: The life of President Johnson.  
> PJPJ: The presidency of President Johnson.  
> LOLO: The life of President Obama.

[Exercise 16](https://aimacode.github.io/aima-exercises/kr-exercises/ex_16/)

This exercise concerns the problem of planning a route for a robot to take from one city to another. The basic action taken by the robot is Go(x,y)Go(x,y), which takes it from city xx to city yy if there is a route between those cities. Road(x,y)Road(x,y) is true if and only if there is a road connecting cities xx and yy; if there is, then Distance(x,y)Distance(x,y) gives the length of the road. See the map on page  for an example. The robot begins in Arad and must reach Bucharest.  
1. Write a suitable logical description of the initial situation of the robot.  
2. Write a suitable logical query whose solutions provide possible paths to the goal.  
3. Write a sentence describing the GoGo action.  
4. Now suppose that the robot consumes fuel at the rate of .02 gallons per mile. The robot starts with 20 gallons of fuel. Augment your representation to include these considerations.  
5. Now suppose some of the cities have gas stations at which the robot can fill its tank. Extend your representation and write all the rules needed to describe gas stations, including the FillupFillup action.

[Exercise 17](https://aimacode.github.io/aima-exercises/kr-exercises/ex_17/)

Investigate ways to extend the event calculus to handle *simultaneous* events. Is it possible to avoid a combinatorial explosion of axioms?

[Exercise 18 (exchange-rates-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_18/)

Construct a representation for exchange rates between currencies that allows for daily fluctuations.

[Exercise 19 (fixed-definition-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_19/)

Define the predicate FixedFixed, where Fixed(Location(x))Fixed(Location(x)) means that the location of object xx is fixed over time.

[Exercise 20](https://aimacode.github.io/aima-exercises/kr-exercises/ex_20/)

Describe the event of trading something for something else. Describe buying as a kind of trading in which one of the objects traded is a sum of money.

[Exercise 21](https://aimacode.github.io/aima-exercises/kr-exercises/ex_21/)

The two preceding exercises assume a fairly primitive notion of ownership. For example, the buyer starts by *owning* the dollar bills. This picture begins to break down when, for example, one’s money is in the bank, because there is no longer any specific collection of dollar bills that one owns. The picture is complicated still further by borrowing, leasing, renting, and bailment. Investigate the various commonsense and legal concepts of ownership, and propose a scheme by which they can be represented formally.

[Exercise 22 (card-on-forehead-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_22/)

(Adapted from [Fagin+al:1995](https://aimacode.github.io/aima-exercises/kr-exercises/).) Consider a game played with a deck of just 8 cards, 4 aces and 4 kings. The three players, Alice, Bob, and Carlos, are dealt two cards each. Without looking at them, they place the cards on their foreheads so that the other players can see them. Then the players take turns either announcing that they know what cards are on their own forehead, thereby winning the game, or saying “I don’t know.” Everyone knows the players are truthful and are perfect at reasoning about beliefs.  
1. Game 1. Alice and Bob have both said “I don’t know.” Carlos sees that Alice has two aces (A-A) and Bob has two kings (K-K). What should Carlos say? (*Hint*: consider all three possible cases for Carlos: A-A, K-K, A-K.)  
2. Describe each step of Game 1 using the notation of modal logic.  
3. Game 2. Carlos, Alice, and Bob all said “I don’t know” on their first turn. Alice holds K-K and Bob holds A-K. What should Carlos say on his second turn?  
4. Game 3. Alice, Carlos, and Bob all say “I don’t know” on their first turn, as does Alice on her second turn. Alice and Bob both hold A-K. What should Carlos say?  
5. Prove that there will always be a winner to this game.

[Exercise 23](https://aimacode.github.io/aima-exercises/kr-exercises/ex_23/)

The assumption of *logical omniscience,* discussed on page , is of course not true of any actual reasoners. Rather, it is an *idealization* of the reasoning process that may be more or less acceptable depending on the applications. Discuss the reasonableness of the assumption for each of the following applications of reasoning about knowledge:  
1. Partial knowledge adversary games, such as card games. Here one player wants to reason about what his opponent knows about the state of the game.  
2. Chess with a clock. Here the player may wish to reason about the limits of his opponent’s or his own ability to find the best move in the time available. For instance, if player A has much more time left than player B, then A will sometimes make a move that greatly complicates the situation, in the hopes of gaining an advantage because he has more time to work out the proper strategy.  
3. A shopping agent in an environment in which there are costs of gathering information.  
4. Reasoning about public key cryptography, which rests on the intractability of certain computational problems.

[Exercise 24](https://aimacode.github.io/aima-exercises/kr-exercises/ex_24/)

The assumption of *logical omniscience,* discussed on page , is of course not true of any actual reasoners. Rather, it is an *idealization* of the reasoning process that may be more or less acceptable depending on the applications. Discuss the reasonableness of the assumption for each of the following applications of reasoning about knowledge:  
1. Partial knowledge adversary games, such as card games. Here one player wants to reason about what his opponent knows about the state of the game.  
2. Chess with a clock. Here the player may wish to reason about the limits of his opponent’s or his own ability to find the best move in the time available. For instance, if player A has much more time left than player B, then A will sometimes make a move that greatly complicates the situation, in the hopes of gaining an advantage because he has more time to work out the proper strategy.  
3. A shopping agent in an environment in which there are costs of gathering information.  
4. Reasoning about public key cryptography, which rests on the intractability of certain computational problems.

[Exercise 25](https://aimacode.github.io/aima-exercises/kr-exercises/ex_25/)

Translate the following description logic expression (from page ) into first-order logic, and comment on the result:

And(Man,AtLeast(3,Son),AtMost(2,Daughter),All(Son,And(Unemployed,Married,All(Spouse,Doctor))),All(Daughter,And(Professor,Fills(Department,Physics,Math))))And(Man,AtLeast(3,Son),AtMost(2,Daughter),All(Son,And(Unemployed,Married,All(Spouse,Doctor))),All(Daughter,And(Professor,Fills(Department,Physics,Math))))

[Exercise 26](https://aimacode.github.io/aima-exercises/kr-exercises/ex_26/)

Recall that inheritance information in semantic networks can be captured logically by suitable implication sentences. This exercise investigates the efficiency of using such sentences for inheritance.  
1. Consider the information in a used-car catalog such as Kelly’s Blue Book—for example, that 1973 Dodge vans are (or perhaps were once) worth 575. Suppose all this information (for 11,000 models) is encoded as logical sentences, as suggested in the chapter. Write down three such sentences, including that for 1973 Dodge vans. How would you use the sentences to find the value of a *particular* car, given a backward-chaining theorem prover such as Prolog?  
2. Compare the time efficiency of the backward-chaining method for solving this problem with the inheritance method used in semantic nets.  
3. Explain how forward chaining allows a logic-based system to solve the same problem efficiently, assuming that the KB contains only the 11,000 sentences about prices.  
4. Describe a situation in which neither forward nor backward chaining on the sentences will allow the price query for an individual car to be handled efficiently.  
5. Can you suggest a solution enabling this type of query to be solved efficiently in all cases in logic systems? *Hint:* Remember that two cars of the same year and model have the same price.)

[Exercise 27 (natural-stupidity-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_27/)

One might suppose that the syntactic distinction between unboxed links and singly boxed links in semantic networks is unnecessary, because singly boxed links are always attached to categories; an inheritance algorithm could simply assume that an unboxed link attached to a category is intended to apply to all members of that category. Show that this argument is fallacious, giving examples of errors that would arise.

[Exercise 28](https://aimacode.github.io/aima-exercises/kr-exercises/ex_28/)

One part of the shopping process that was not covered in this chapter is checking for compatibility between items. For example, if a digital camera is ordered, what accessory batteries, memory cards, and cases are compatible with the camera? Write a knowledge base that can determine the compatibility of a set of items and suggest replacements or additional items if the shopper makes a choice that is not compatible. The knowledge base should works with at least one line of products and extend easily to other lines.

[Exercise 29 (shopping-grammar-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_29/)

A complete solution to the problem of inexact matches to the buyer’s description in shopping is very difficult and requires a full array of natural language processing and information retrieval techniques. (See Chapters [Natural Language for Communication](https://aimacode.github.io/aima-exercises/nlp1-exercises/) and [Natural Language Processing](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/).) One small step is to allow the user to specify minimum and maximum values for various attributes. The buyer must use the following grammar for product descriptions:

Description→Category [Connector Modifier]∗Description→Category [Connector Modifier]∗

Connector→"with" |"and"|","Connector→"with" |"and"|","

Modifier→Attribute | Attribute Op ValueModifier→Attribute | Attribute Op Value

Op→"="|">"|"<"Op→"="|">"|"<"

Here, CategoryCategory names a product category, AttributeAttribute is some feature such as “CPU” or “price,” and ValueValue is the target value for the attribute. So the query “computer with at least a 2.5 GHz CPU for under 500” must be re-expressed as “computer with CPU >> 2.5 GHz and price << 500.” Implement a shopping agent that accepts descriptions in this language.

[Exercise 30 (buying-exercise)](https://aimacode.github.io/aima-exercises/kr-exercises/ex_30/)

Our description of Internet shopping omitted the all-important step of actually *buying* the product. Provide a formal logical description of buying, using event calculus. That is, define the sequence of events that occurs when a buyer submits a credit-card purchase and then eventually gets billed and receives the product.

**13. Quantifying Uncertainity**

[Exercise 1](https://aimacode.github.io/aima-exercises/probability-exercises/ex_1/)

Show from first principles that P(ab∧a)=1P(ab∧a)=1.

[Exercise 2 (sum-to-1-exercise)](https://aimacode.github.io/aima-exercises/probability-exercises/ex_2/)

Using the axioms of probability, prove that any probability distribution on a discrete random variable must sum to 1.

[Exercise 3](https://aimacode.github.io/aima-exercises/probability-exercises/ex_3/)

For each of the following statements, either prove it is true or give a counterexample.  
1. If P(ab,c)=P(ba,c)P(ab,c)=P(ba,c), then P(ac)=P(bc)P(ac)=P(bc)  
2. If P(ab,c)=P(a)P(ab,c)=P(a), then P(bc)=P(b)P(bc)=P(b)  
3. If P(ab)=P(a)P(ab)=P(a), then P(ab,c)=P(ac)P(ab,c)=P(ac)

[Exercise 4](https://aimacode.github.io/aima-exercises/probability-exercises/ex_4/)

Would it be rational for an agent to hold the three beliefs P(A)=0.4P(A)=0.4, P(B)=0.3P(B)=0.3, and P(A∨B)=0.5P(A∨B)=0.5? If so, what range of probabilities would be rational for the agent to hold for A∧BA∧B? Make up a table like the one in Figure [13.2](https://aimacode.github.io/aima-exercises/figures/de-finetti-table.png), and show how it supports your argument about rationality. Then draw another version of the table where P(A∨B)=0.7P(A∨B)=0.7. Explain why it is rational to have this probability, even though the table shows one case that is a loss and three that just break even. (*Hint:* what is Agent 1 committed to about the probability of each of the four cases, especially the case that is a loss?)

[Exercise 5 (exclusive-exhaustive-exercise)](https://aimacode.github.io/aima-exercises/probability-exercises/ex_5/)

This question deals with the properties of possible worlds, defined on page  as assignments to all random variables. We will work with propositions that correspond to exactly one possible world because they pin down the assignments of all the variables. In probability theory, such propositions are called **atomic event**. For example, with Boolean variables X1X1, X2X2, X3X3, the proposition x1∧¬x2∧¬x3x1∧¬x2∧¬x3 fixes the assignment of the variables; in the language of propositional logic, we would say it has exactly one model.  
1. Prove, for the case of nn Boolean variables, that any two distinct atomic events are mutually exclusive; that is, their conjunction is equivalent to falsefalse.  
2. Prove that the disjunction of all possible atomic events is logically equivalent to truetrue.  
3. Prove that any proposition is logically equivalent to the disjunction of the atomic events that entail its truth.

[Exercise 6 (inclusion-exclusion-exercise)](https://aimacode.github.io/aima-exercises/probability-exercises/ex_6/)

Prove Equation () from Equations  and (.

[Exercise 7](https://aimacode.github.io/aima-exercises/probability-exercises/ex_7/)

Consider the set of all possible five-card poker hands dealt fairly from a standard deck of fifty-two cards.  
1. How many atomic events are there in the joint probability distribution (i.e., how many five-card hands are there)?  
2. What is the probability of each atomic event?  
3. What is the probability of being dealt a royal straight flush? Four of a kind?

[Exercise 8](https://aimacode.github.io/aima-exercises/probability-exercises/ex_8/)

Given the full joint distribution shown in Figure [13.3](https://aimacode.github.io/aima-exercises/figures/dentist-joint-table.png), calculate the following:  
1. P(toothache)P(toothache).  
2. P(Cavity)P(Cavity).  
3. P(Toothachecavity)P(Toothachecavity).  
4. P(Cavitytoothache∨catch)P(Cavitytoothache∨catch).

[Exercise 9](https://aimacode.github.io/aima-exercises/probability-exercises/ex_9/)

Given the full joint distribution shown in Figure [13.3](https://aimacode.github.io/aima-exercises/figures/dentist-joint-table.png), calculate the following:  
1. P(toothache)P(toothache).  
2. P(Catch)P(Catch).  
3. P(Cavitycatch)P(Cavitycatch).  
4. P(Cavitytoothache∨catch)P(Cavitytoothache∨catch).

[Exercise 10 (unfinished-game-exercise)](https://aimacode.github.io/aima-exercises/probability-exercises/ex_10/)

In his letter of August 24, 1654, Pascal was trying to show how a pot of money should be allocated when a gambling game must end prematurely. Imagine a game where each turn consists of the roll of a die, player *E* gets a point when the die is even, and player *O* gets a point when the die is odd. The first player to get 7 points wins the pot. Suppose the game is interrupted with *E* leading 4–2. How should the money be fairly split in this case? What is the general formula? (Fermat and Pascal made several errors before solving the problem, but you should be able to get it right the first time.)

[Exercise 11](https://aimacode.github.io/aima-exercises/probability-exercises/ex_11/)

Deciding to put probability theory to good use, we encounter a slot machine with three independent wheels, each producing one of the four symbols bar, bell, lemon, or cherry with equal probability. The slot machine has the following payout scheme for a bet of 1 coin (where “?” denotes that we don’t care what comes up for that wheel):  
> bar/bar/bar pays 20 coins  
> bell/bell/bell pays 15 coins  
> lemon/lemon/lemon pays 5 coins  
> cherry/cherry/cherry pays 3 coins  
> cherry/cherry/? pays 2 coins  
> cherry/?/? pays 1 coin  
1. Compute the expected “payback” percentage of the machine. In other words, for each coin played, what is the expected coin return?  
2. Compute the probability that playing the slot machine once will result in a win.  
3. Estimate the mean and median number of plays you can expect to make until you go broke, if you start with 10 coins. You can run a simulation to estimate this, rather than trying to compute an exact answer.

[Exercise 12](https://aimacode.github.io/aima-exercises/probability-exercises/ex_12/)

Deciding to put probability theory to good use, we encounter a slot machine with three independent wheels, each producing one of the four symbols bar, bell, lemon, or cherry with equal probability. The slot machine has the following payout scheme for a bet of 1 coin (where “?” denotes that we don’t care what comes up for that wheel):  
> bar/bar/bar pays 20 coins  
> bell/bell/bell pays 15 coins  
> lemon/lemon/lemon pays 5 coins  
> cherry/cherry/cherry pays 3 coins  
> cherry/cherry/? pays 2 coins  
> cherry/?/? pays 1 coin  
1. Compute the expected “payback” percentage of the machine. In other words, for each coin played, what is the expected coin return?  
2. Compute the probability that playing the slot machine once will result in a win.  
3. Estimate the mean and median number of plays you can expect to make until you go broke, if you start with 10 coins. You can run a simulation to estimate this, rather than trying to compute an exact answer.

[Exercise 13](https://aimacode.github.io/aima-exercises/probability-exercises/ex_13/)

We wish to transmit an nn-bit message to a receiving agent. The bits in the message are independently corrupted (flipped) during transmission with ϵϵ probability each. With an extra parity bit sent along with the original information, a message can be corrected by the receiver if at most one bit in the entire message (including the parity bit) has been corrupted. Suppose we want to ensure that the correct message is received with probability at least 1−δ1−δ. What is the maximum feasible value of nn? Calculate this value for the case ϵ=0.001ϵ=0.001, δ=0.01δ=0.01.

[Exercise 14](https://aimacode.github.io/aima-exercises/probability-exercises/ex_14/)

We wish to transmit an nn-bit message to a receiving agent. The bits in the message are independently corrupted (flipped) during transmission with ϵϵ probability each. With an extra parity bit sent along with the original information, a message can be corrected by the receiver if at most one bit in the entire message (including the parity bit) has been corrupted. Suppose we want to ensure that the correct message is received with probability at least 1−δ1−δ. What is the maximum feasible value of nn? Calculate this value for the case ϵ0.002ϵ0.002, δ0.01δ0.01.

[Exercise 15 (independence-exercise)](https://aimacode.github.io/aima-exercises/probability-exercises/ex_15/)

Show that the three forms of independence in Equation () are equivalent.

[Exercise 16](https://aimacode.github.io/aima-exercises/probability-exercises/ex_16/)

Consider two medical tests, A and B, for a virus. Test A is 95% effective at recognizing the virus when it is present, but has a 10% false positive rate (indicating that the virus is present, when it is not). Test B is 90% effective at recognizing the virus, but has a 5% false positive rate. The two tests use independent methods of identifying the virus. The virus is carried by 1% of all people. Say that a person is tested for the virus using only one of the tests, and that test comes back positive for carrying the virus. Which test returning positive is more indicative of someone really carrying the virus? Justify your answer mathematically.

[Exercise 17](https://aimacode.github.io/aima-exercises/probability-exercises/ex_17/)

Suppose you are given a coin that lands headsheads with probability xx and tailstails with probability 1−x1−x. Are the outcomes of successive flips of the coin independent of each other given that you know the value of xx? Are the outcomes of successive flips of the coin independent of each other if you do *not* know the value of xx? Justify your answer.

[Exercise 18](https://aimacode.github.io/aima-exercises/probability-exercises/ex_18/)

After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a serious disease and that the test is 99% accurate (i.e., the probability of testing positive when you do have the disease is 0.99, as is the probability of testing negative when you don’t have the disease). The good news is that this is a rare disease, striking only 1 in 10,000 people of your age. Why is it good news that the disease is rare? What are the chances that you actually have the disease?

[Exercise 19](https://aimacode.github.io/aima-exercises/probability-exercises/ex_19/)

After your yearly checkup, the doctor has bad news and good news. The bad news is that you tested positive for a serious disease and that the test is 99% accurate (i.e., the probability of testing positive when you do have the disease is 0.99, as is the probability of testing negative when you don’t have the disease). The good news is that this is a rare disease, striking only 1 in 100,000 people of your age. Why is it good news that the disease is rare? What are the chances that you actually have the disease?

[Exercise 20 (conditional-bayes-exercise)](https://aimacode.github.io/aima-exercises/probability-exercises/ex_20/)

It is quite often useful to consider the effect of some specific propositions in the context of some general background evidence that remains fixed, rather than in the complete absence of information. The following questions ask you to prove more general versions of the product rule and Bayes’ rule, with respect to some background evidence ee:  
1. Prove the conditionalized version of the general product rule:

P(X,Ye)=P(XY,e)P(Ye) .P(X,Ye)=P(XY,e)P(Ye) .

2. Prove the conditionalized version of Bayes’ rule in Equation ().

[Exercise 21 (pv-xyz-exercise)](https://aimacode.github.io/aima-exercises/probability-exercises/ex_21/)

Show that the statement of conditional independence

P(X,Y|Z)=P(X|Z)P(Y|Z)P(X,Y|Z)=P(X|Z)P(Y|Z)

is equivalent to each of the statements

P(X|Y,Z)=P(X|Z)andP(Y|X,Z)=P(Y|Z) .P(X|Y,Z)=P(X|Z)andP(Y|X,Z)=P(Y|Z) .

[Exercise 22](https://aimacode.github.io/aima-exercises/probability-exercises/ex_22/)

Suppose you are given a bag containing nn unbiased coins. You are told that n−1n−1 of these coins are normal, with heads on one side and tails on the other, whereas one coin is a fake, with heads on both sides.  
1. Suppose you reach into the bag, pick out a coin at random, flip it, and get a head. What is the (conditional) probability that the coin you chose is the fake coin?  
2. Suppose you continue flipping the coin for a total of kk times after picking it and see kk heads. Now what is the conditional probability that you picked the fake coin?  
3. Suppose you wanted to decide whether the chosen coin was fake by flipping it kk times. The decision procedure returns fakefake if all kk flips come up heads; otherwise it returns normalnormal. What is the (unconditional) probability that this procedure makes an error?

[Exercise 23 (normalization-exercise)](https://aimacode.github.io/aima-exercises/probability-exercises/ex_23/)

In this exercise, you will complete the normalization calculation for the meningitis example. First, make up a suitable value for P(s¬m)P(s¬m), and use it to calculate unnormalized values for P(ms)P(ms) and P(¬ms)P(¬ms) (i.e., ignoring the P(s)P(s) term in the Bayes’ rule expression, Equation (). Now normalize these values so that they add to 1.

[Exercise 24](https://aimacode.github.io/aima-exercises/probability-exercises/ex_24/)

This exercise investigates the way in which conditional independence relationships affect the amount of information needed for probabilistic calculations.  
1. Suppose we wish to calculate P(he1,e2)P(he1,e2) and we have no conditional independence information. Which of the following sets of numbers are sufficient for the calculation?  
1. P(E1,E2)P(E1,E2), P(H)P(H), P(E1H)P(E1H), P(E2H)P(E2H) 2. P(E1,E2)P(E1,E2), P(H)P(H), P(E1,E2H)P(E1,E2H)  
3. P(H)P(H), P(E1H)P(E1H), P(E2H)P(E2H)  
2. Suppose we know that P(E1H,E2)=P(E1H)P(E1H,E2)=P(E1H) for all values of HH, E1E1, E2E2. Now which of the three sets are sufficient?

[Exercise 25](https://aimacode.github.io/aima-exercises/probability-exercises/ex_25/)

Let XX, YY, ZZ be Boolean random variables. Label the eight entries in the joint distribution P(X,Y,Z)P(X,Y,Z) as aa through hh. Express the statement that XX and YY are conditionally independent given ZZ, as a set of equations relating aa through hh. How many *nonredundant*equations are there?

[Exercise 26](https://aimacode.github.io/aima-exercises/probability-exercises/ex_26/)

(Adapted from Pearl [[Pearl:1988](https://aimacode.github.io/aima-exercises/probability-exercises/)].) Suppose you are a witness to a nighttime hit-and-run accident involving a taxi in Athens. All taxis in Athens are blue or green. You swear, under oath, that the taxi was blue. Extensive testing shows that, under the dim lighting conditions, discrimination between blue and green is 75% reliable.  
1. Is it possible to calculate the most likely color for the taxi? (\*Hint:\* distinguish carefully between the proposition that the taxi \*is\* blue and the proposition that it \*appears\* blue.)  
2. What if you know that 9 out of 10 Athenian taxis are green?

[Exercise 27](https://aimacode.github.io/aima-exercises/probability-exercises/ex_27/)

Write out a general algorithm for answering queries of the form P(Causee)P(Causee), using a naive Bayes distribution. Assume that the evidence ee may assign values to *any subset* of the effect variables.

[Exercise 28 (naive-bayes-retrieval-exercise)](https://aimacode.github.io/aima-exercises/probability-exercises/ex_28/)

Text categorization is the task of assigning a given document to one of a fixed set of categories on the basis of the text it contains. Naive Bayes models are often used for this task. In these models, the query variable is the document category, and the “effect” variables are the presence or absence of each word in the language; the assumption is that words occur independently in documents, with frequencies determined by the document category.  
1. Explain precisely how such a model can be constructed, given as “training data” a set of documents that have been assigned to categories.  
2. Explain precisely how to categorize a new document.  
3. Is the conditional independence assumption reasonable? Discuss.

[Exercise 29](https://aimacode.github.io/aima-exercises/probability-exercises/ex_29/)

In our analysis of the wumpus world, we used the fact that each square contains a pit with probability 0.2, independently of the contents of the other squares. Suppose instead that exactly N/5N/5 pits are scattered at random among the NN squares other than [1,1]. Are the variables Pi,jPi,j and Pk,lPk,l still independent? What is the joint distribution P(P1,1,…,P4,4)P(P1,1,…,P4,4) now? Redo the calculation for the probabilities of pits in [1,3] and [2,2].

[Exercise 30](https://aimacode.github.io/aima-exercises/probability-exercises/ex_30/)

Redo the probability calculation for pits in [1,3] and [2,2], assuming that each square contains a pit with probability 0.01, independent of the other squares. What can you say about the relative performance of a logical versus a probabilistic agent in this case?

[Exercise 31](https://aimacode.github.io/aima-exercises/probability-exercises/ex_31/)

Implement a hybrid probabilistic agent for the wumpus world, based on the hybrid agent in Figure [7.20](https://aimacode.github.io/aima-exercises/figures/hybrid-wumpus-agent-algorithm.png) and the probabilistic inference procedure outlined in this chapter.

[Artificial Intelligence](https://aimacode.github.io/aima-exercises/) AIMA Exercises

窗体顶端

窗体底端

14. Probabilistic Reasoning

[Exercise 1](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_1/)

We have a bag of three biased coins aa, bb, and cc with probabilities of coming up heads of 20%, 60%, and 80%, respectively. One coin is drawn randomly from the bag (with equal likelihood of drawing each of the three coins), and then the coin is flipped three times to generate the outcomes X1X1, X2X2, and X3X3.  
1. Draw the Bayesian network corresponding to this setup and define the necessary CPTs.  
2. Calculate which coin was most likely to have been drawn from the bag if the observed flips come out heads twice and tails once.

[Exercise 2](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_2/)

We have a bag of three biased coins aa, bb, and cc with probabilities of coming up heads of 30%, 60%, and 75%, respectively. One coin is drawn randomly from the bag (with equal likelihood of drawing each of the three coins), and then the coin is flipped three times to generate the outcomes X1X1, X2X2, and X3X3.  
1. Draw the Bayesian network corresponding to this setup and define the necessary CPTs.  
2. Calculate which coin was most likely to have been drawn from the bag if the observed flips come out heads twice and tails once.

[Exercise 3(cpt-equivalence-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_3/)

Equation ([parameter-joint-repn-equation](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/) on page  defines the joint distribution represented by a Bayesian network in terms of the parameters θ(XiParents(Xi))θ(XiParents(Xi)). This exercise asks you to derive the equivalence between the parameters and the conditional probabilities  P(XiParents(Xi)) P(XiParents(Xi)) from this definition.  
1. Consider a simple network X→Y→ZX→Y→Z with three Boolean variables. Use Equations ( and ( (pages [conditional-probability-equation](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/) and [marginalization-equation](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/)) to express the conditional probability P(zy)P(zy) as the ratio of two sums, each over entries in the joint distribution P(X,Y,Z)P(X,Y,Z).  
2. Now use Equation ( to write this expression in terms of the network parameters θ(X)θ(X), θ(YX)θ(YX), and θ(ZY)θ(ZY).  
3. Next, expand out the summations in your expression from part (b), writing out explicitly the terms for the true and false values of each summed variable. Assuming that all network parameters satisfy the constraint ∑xiθ(xiparents(Xi))1∑xiθ(xiparents(Xi))1, show that the resulting expression reduces to θ(zy)θ(zy).  
4. Generalize this derivation to show that θ(XiParents(Xi))=P(XiParents(Xi))θ(XiParents(Xi))=P(XiParents(Xi)) for any Bayesian network.

[Exercise 4](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_4/)

The **arc reversal** operation of in a Bayesian network allows us to change the direction of an arc X→YX→Y while preserving the joint probability distribution that the network represents [Shachter:1986](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/). Arc reversal may require introducing new arcs: all the parents of XX also become parents of YY, and all parents of YY also become parents of XX.  
1. Assume that XX and YY start with mm and nn parents, respectively, and that all variables have kk values. By calculating the change in size for the CPTs of XX and YY, show that the total number of parameters in the network cannot decrease during arc reversal. (*Hint*: the parents of XX and YY need not be disjoint.)  
2. Under what circumstances can the total number remain constant?  
3. Let the parents of XX be U∪VU∪V and the parents of YY be V∪WV∪W, where UU and WW are disjoint. The formulas for the new CPTs after arc reversal are as follows:

P(Y|U,V,W)P(X|U,V,W,Y)==∑xP(Y|V,W,x)P(x|U,V)P(Y|X,V,W)P(X|U,V)/P(Y|U,V,W) .P(Y|U,V,W)=∑xP(Y|V,W,x)P(x|U,V)P(X|U,V,W,Y)=P(Y|X,V,W)P(X|U,V)/P(Y|U,V,W) .

Prove that the new network expresses the same joint distribution over all variables as the original network.

[Exercise 5](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_5/)

Consider the Bayesian network in Figure [burglary-figure.](https://aimacode.github.io/aima-exercises/figures/burglary-figure.png)  
1. If no evidence is observed, are BurglaryBurglary and EarthquakeEarthquake independent? Prove this from the numerical semantics and from the topological semantics.  
2. If we observe AlarmtrueAlarmtrue, are BurglaryBurglary and EarthquakeEarthquake independent? Justify your answer by calculating whether the probabilities involved satisfy the definition of conditional independence.

[Exercise 6](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_6/)

Suppose that in a Bayesian network containing an unobserved variable YY, all the variables in the Markov blanket MB(Y)MB(Y) have been observed.  
1. Prove that removing the node YY from the network will not affect the posterior distribution for any other unobserved variable in the network.  
2. Discuss whether we can remove YY if we are planning to use (i) rejection sampling and (ii) likelihood weighting.

**Three possible structures for a Bayesian network describing genetic inheritance of handedness.**

[Exercise 7 (handedness-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_7/)

Let HxHx be a random variable denoting the handedness of an individual xx, with possible values ll or rr. A common hypothesis is that left- or right-handedness is inherited by a simple mechanism; that is, perhaps there is a gene GxGx, also with values ll or rr, and perhaps actual handedness turns out mostly the same (with some probability ss) as the gene an individual possesses. Furthermore, perhaps the gene itself is equally likely to be inherited from either of an individual’s parents, with a small nonzero probability mm of a random mutation flipping the handedness.  
1. Which of the three networks in Figure [handedness-figure](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/#handedness-figure) claim that P(Gfather,Gmother,Gchild)=P(Gfather)P(Gmother)P(Gchild)P(Gfather,Gmother,Gchild)=P(Gfather)P(Gmother)P(Gchild)?  
2. Which of the three networks make independence claims that are consistent with the hypothesis about the inheritance of handedness?  
3. Which of the three networks is the best description of the hypothesis?  
4. Write down the CPT for the GchildGchild node in network (a), in terms of ss and mm.  
5. Suppose that P(Gfatherl)=P(Gmotherl)=qP(Gfatherl)=P(Gmotherl)=q. In network (a), derive an expression for P(Gchildl)P(Gchildl) in terms of mm and qq only, by conditioning on its parent nodes.  
6. Under conditions of genetic equilibrium, we expect the distribution of genes to be the same across generations. Use this to calculate the value of qq, and, given what you know about handedness in humans, explain why the hypothesis described at the beginning of this question must be wrong.

[Exercise 8 (markov-blanket-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_8/)

The **Markov blanket** of a variable is defined on page [markov-blanket-page](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/). Prove that a variable is independent of all other variables in the network, given its Markov blanket and derive Equation () (page ).

**A Bayesian network describing some features of a car's electrical system and engine. Each variable is Boolean, and the *true* value indicates that the corresponding aspect of the vehicle is in working order.**

[Exercise 9](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_9/)

Consider the network for car diagnosis shown in Figure [car-starts-figure](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/#car-starts-figure)  
. 1. Extend the network with the Boolean variables IcyWeatherIcyWeather and StarterMotorStarterMotor.  
2. Give reasonable conditional probability tables for all the nodes.  
3. How many independent values are contained in the joint probability distribution for eight Boolean nodes, assuming that no conditional independence relations are known to hold among them?  
4. How many independent probability values do your network tables contain?  
5. The conditional distribution for StartsStarts could be described as a **noisy-AND** distribution. Define this family in general and relate it to the noisy-OR distribution.

[Exercise 10](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_10/)

Consider a simple Bayesian network with root variables ColdCold, FluFlu, and MalariaMalaria and child variable FeverFever, with a noisy-OR conditional distribution for FeverFever as described in Section . By adding appropriate auxiliary variables for inhibition events and fever-inducing events, construct an equivalent Bayesian network whose CPTs (except for root variables) are deterministic. Define the CPTs and prove equivalence.

[Exercise 11 (LG-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_11/)

Consider the family of linear Gaussian networks, as defined on page   
. 1. In a two-variable network, let X1X1 be the parent of X2X2, let X1X1 have a Gaussian prior, and let P(X2X1)P(X2X1) be a linear Gaussian distribution. Show that the joint distribution P(X1,X2)P(X1,X2) is a multivariate Gaussian, and calculate its covariance matrix.  
2. Prove by induction that the joint distribution for a general linear Gaussian network on X1,…,XnX1,…,Xn is also a multivariate Gaussian.

[Exercise 12 (multivalued-probit-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_12/)

The probit distribution defined on page  describes the probability distribution for a Boolean child, given a single continuous parent.  
1. How might the definition be extended to cover multiple continuous parents?  
2. How might it be extended to handle a *multivalued* child variable? Consider both cases where the child’s values are ordered (as in selecting a gear while driving, depending on speed, slope, desired acceleration, etc.) and cases where they are unordered (as in selecting bus, train, or car to get to work). (*Hint*: Consider ways to divide the possible values into two sets, to mimic a Boolean variable.)

[Exercise 13](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_13/)

In your local nuclear power station, there is an alarm that senses when a temperature gauge exceeds a given threshold. The gauge measures the temperature of the core. Consider the Boolean variables AA (alarm sounds), FAFA (alarm is faulty), and FGFG (gauge is faulty) and the multivalued nodes GG (gauge reading) and TT (actual core temperature).  
1. Draw a Bayesian network for this domain, given that the gauge is more likely to fail when the core temperature gets too high.  
2. Is your network a polytree? Why or why not?  
3. Suppose there are just two possible actual and measured temperatures, normal and high; the probability that the gauge gives the correct temperature is xx when it is working, but yy when it is faulty. Give the conditional probability table associated with GG.  
4. Suppose the alarm works correctly unless it is faulty, in which case it never sounds. Give the conditional probability table associated with AA.  
5. Suppose the alarm and gauge are working and the alarm sounds. Calculate an expression for the probability that the temperature of the core is too high, in terms of the various conditional probabilities in the network.

[Exercise 14 (telescope-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_14/)

Two astronomers in different parts of the world make measurements M1M1 and M2M2 of the number of stars NN in some small region of the sky, using their telescopes. Normally, there is a small possibility ee of error by up to one star in each direction. Each telescope can also (with a much smaller probability ff) be badly out of focus (events F1F1 and F2F2), in which case the scientist will undercount by three or more stars (or if NN is less than 3, fail to detect any stars at all). Consider the three networks shown in Figure [telescope-nets-figure](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/#telescope-nets-figure).  
1. Which of these Bayesian networks are correct (but not necessarily efficient) representations of the preceding information?  
2. Which is the best network? Explain.  
3. Write out a conditional distribution for P(M1N)P(M1N), for the case where N{1,2,3}N{1,2,3} and M1{0,1,2,3,4}M1{0,1,2,3,4}. Each entry in the conditional distribution should be expressed as a function of the parameters ee and/or ff.  
4. Suppose M11M11 and M23M23. What are the *possible* numbers of stars if you assume no prior constraint on the values of NN?  
5. What is the *most likely* number of stars, given these observations? Explain how to compute this, or if it is not possible to compute, explain what additional information is needed and how it would affect the result.

[Exercise 15](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_15/)

Consider the network shown in Figure [telescope-nets-figure](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/#telescope-nets-figure)(ii), and assume that the two telescopes work identically. N{1,2,3}N{1,2,3} and M1,M2{0,1,2,3,4}M1,M2{0,1,2,3,4}, with the symbolic CPTs as described in Exercise [14.14](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/undefined). Using the enumeration algorithm (Figure [14.22](https://aimacode.github.io/aima-exercises/figures/enumeration-algorithm.png) on page ), calculate the probability distribution P(NM12,M22)P(NM12,M22).

**Three possible networks for the telescope problem.**

[Exercise 16](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_16/)

Consider the Bayes net shown in Figure [politics-figure](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/#politics-figure)  
. 1. Which of the following are asserted by the network *structure*?  
1. P(B,I,M)=P(B)P(I)P(M)P(B,I,M)=P(B)P(I)P(M).  
2. P(J|G)=P(J|G,I)P(J|G)=P(J|G,I).  
3. P(M|G,B,I)=P(M|G,B,I,J)P(M|G,B,I)=P(M|G,B,I,J).  
2. Calculate the value of P(b,i,¬m,g,j)P(b,i,¬m,g,j).  
3. Calculate the probability that someone goes to jail given that they broke the law, have been indicted, and face a politically motivated prosecutor.  
4. A **context-specific independence** (see page ) allows a variable to be independent of some of its parents given certain values of others. In addition to the usual conditional independences given by the graph structure, what context-specific independences exist in the Bayes net in Figure [politics-figure](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/#politics-figure)?  
5. Suppose we want to add the variable P=PresidentialPardonP=PresidentialPardon to the network; draw the new network and briefly explain any links you add.

**A simple Bayes net with Boolean variables B = {BrokeElectionLaw}, I = {Indicted}, M = {PoliticallyMotivatedProsecutor}, G= {FoundGuilty}, J = {Jailed}.**

[Exercise 17](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_17/)

Consider the Bayes net shown in Figure [politics-figure](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/#politics-figure)  
. 1. Which of the following are asserted by the network *structure*?  
1. P(B,I,M)=P(B)P(I)P(M)P(B,I,M)=P(B)P(I)P(M).  
2. P(J|G)=P(J|G,I)P(J|G)=P(J|G,I).  
3. P(M|G,B,I)=P(M|G,B,I,J)P(M|G,B,I)=P(M|G,B,I,J).  
2. Calculate the value of P(b,i,¬m,g,j)P(b,i,¬m,g,j).  
3. Calculate the probability that someone goes to jail given that they broke the law, have been indicted, and face a politically motivated prosecutor.  
4. A **context-specific independence** (see page ) allows a variable to be independent of some of its parents given certain values of others. In addition to the usual conditional independences given by the graph structure, what context-specific independences exist in the Bayes net in Figure [politics-figure](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/#politics-figure)?  
5. Suppose we want to add the variable P=PresidentialPardonP=PresidentialPardon to the network; draw the new network and briefly explain any links you add.

[Exercise 18 (VE-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_18/)

Consider the variable elimination algorithm in Figure [14.11](https://aimacode.github.io/aima-exercises/figures/elimination-ask-algorithm.png) (page ).  
1. Section  applies variable elimination to the query

P(BurglaryJohnCallstrue,MaryCallstrue) .P(BurglaryJohnCallstrue,MaryCallstrue) .

Perform the calculations indicated and check that the answer is correct.  
2. Count the number of arithmetic operations performed, and compare it with the number performed by the enumeration algorithm.  
3. Suppose a network has the form of a *chain*: a sequence of Boolean variables X1,…,XnX1,…,Xn where Parents(Xi){Xi−1}Parents(Xi){Xi−1} for i2,…,ni2,…,n. What is the complexity of computing P(X1Xntrue)P(X1Xntrue) using enumeration? Using variable elimination?  
4. Prove that the complexity of running variable elimination on a polytree network is linear in the size of the tree for any variable ordering consistent with the network structure.

[Exercise 19 (bn-complexity-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_19/)

Investigate the complexity of exact inference in general Bayesian networks:  
1. Prove that any 3-SAT problem can be reduced to exact inference in a Bayesian network constructed to represent the particular problem and hence that exact inference is NP-hard. (*Hint*: Consider a network with one variable for each proposition symbol, one for each clause, and one for the conjunction of clauses.)  
2. The problem of counting the number of satisfying assignments for a 3-SAT problem is \#P-complete. Show that exact inference is at least as hard as this.

[Exercise 20 (primitive-sampling-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_20/)

Consider the problem of generating a random sample from a specified distribution on a single variable. Assume you have a random number generator that returns a random number uniformly distributed between 0 and 1.  
1. Let XX be a discrete variable with P(Xxi)piP(Xxi)pi for i{1,…,k}i{1,…,k}. The **cumulative distribution** of XX gives the probability that X{x1,…,xj}X{x1,…,xj} for each possible jj. (See also Appendix [math-appendix].) Explain how to calculate the cumulative distribution in O(k)O(k) time and how to generate a single sample of XX from it. Can the latter be done in less than O(k)O(k) time?  
2. Now suppose we want to generate NN samples of XX, where N≫kN≫k. Explain how to do this with an expected run time per sample that is *constant* (i.e., independent of kk).  
3. Now consider a continuous-valued variable with a parameterized distribution (e.g., Gaussian). How can samples be generated from such a distribution?  
4. Suppose you want to query a continuous-valued variable and you are using a sampling algorithm such as LIKELIHOODWEIGHTING to do the inference. How would you have to modify the query-answering process?

[Exercise 21](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_21/)

Consider the query P(RainSprinklertrue,WetGrasstrue)P(RainSprinklertrue,WetGrasstrue) in Figure [14.12](https://aimacode.github.io/aima-exercises/figures/rain-clustering-figure.png)(a) (page ) and how Gibbs sampling can answer it.  
1. How many states does the Markov chain have?  
2. Calculate the **transition matrix** QQ containing q(yq(y →→ y′)y′) for all yy, y′y′.  
3. What does  Q2 Q2, the square of the transition matrix, represent?  
4. What about QnQn as n→∞n→∞?  
5. Explain how to do probabilistic inference in Bayesian networks, assuming that QnQn is available. Is this a practical way to do inference?

[Exercise 22 (gibbs-proof-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_22/)

This exercise explores the stationary distribution for Gibbs sampling methods.  
1. The convex composition [α,q1;1−α,q2][α,q1;1−α,q2] of q1q1 and q2q2 is a transition probability distribution that first chooses one of q1q1 and q2q2 with probabilities αα and 1−α1−α, respectively, and then applies whichever is chosen. Prove that if q1q1 and q2q2 are in detailed balance with ππ, then their convex composition is also in detailed balance with ππ. (*Note*: this result justifies a variant of GIBBS-ASK in which variables are chosen at random rather than sampled in a fixed sequence.)  
2. Prove that if each of q1q1 and q2q2 has ππ as its stationary distribution, then the sequential composition qq1∘q2qq1∘q2 also has ππ as its stationary distribution.

[Exercise 23 (MH-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_23/)

The **Metropolis--Hastings** algorithm is a member of the MCMC family; as such, it is designed to generate samples xx (eventually) according to target probabilities π(x)π(x). (Typically we are interested in sampling from π(x)P(xe)π(x)P(xe).) Like simulated annealing, Metropolis–Hastings operates in two stages. First, it samples a new state x'x' from a **proposal distribution** q(x'x)q(x'x), given the current state xx. Then, it probabilistically accepts or rejects x'x' according to the **acceptance probability**

α(x'x)=min (1,π(x')q(xx')π(x)q(x'x)) .α(x'x)=min (1,π(x')q(xx')π(x)q(x'x)) .

If the proposal is rejected, the state remains at xx.  
1. Consider an ordinary Gibbs sampling step for a specific variable XiXi. Show that this step, considered as a proposal, is guaranteed to be accepted by Metropolis–Hastings. (Hence, Gibbs sampling is a special case of Metropolis–Hastings.)  
2. Show that the two-step process above, viewed as a transition probability distribution, is in detailed balance with ππ.

[Exercise 24 (soccer-rpm-exercise)](https://aimacode.github.io/aima-exercises/bayes-nets-exercises/ex_24/)

Three soccer teams AA, BB, and CC, play each other once. Each match is between two teams, and can be won, drawn, or lost. Each team has a fixed, unknown degree of quality—an integer ranging from 0 to 3—and the outcome of a match depends probabilistically on the difference in quality between the two teams.  
1. Construct a relational probability model to describe this domain, and suggest numerical values for all the necessary probability distributions.  
2. Construct the equivalent Bayesian network for the three matches.  
3. Suppose that in the first two matches AA beats BB and draws with CC. Using an exact inference algorithm of your choice, compute the posterior distribution for the outcome of the third match.  
4. Suppose there are nn teams in the league and we have the results for all but the last match. How does the complexity of predicting the last game vary with nn?  
5. Investigate the application of MCMC to this problem. How quickly does it converge in practice and how well does it scale?

15. Probabilistic Reasoning over Time

[Exercise 1 (state-augmentation-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_1/)

Show that any second-order Markov process can be rewritten as a first-order Markov process with an augmented set of state variables. Can this always be done *parsimoniously*, i.e., without increasing the number of parameters needed to specify the transition model?

[Exercise 2 (markov-convergence-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_2/)

In this exercise, we examine what happens to the probabilities in the umbrella world in the limit of long time sequences.  
1. Suppose we observe an unending sequence of days on which the umbrella appears. Show that, as the days go by, the probability of rain on the current day increases monotonically toward a fixed point. Calculate this fixed point.  
2. Now consider *forecasting* further and further into the future, given just the first two umbrella observations. First, compute the probability P(r2+k|u1,u2)P(r2+k|u1,u2) for k=1…20k=1…20 and plot the results. You should see that the probability converges towards a fixed point. Prove that the exact value of this fixed point is 0.5.

[Exercise 3 (island-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_3/)

This exercise develops a space-efficient variant of the forward–backward algorithm described in Figure [8.2](https://aimacode.github.io/aima-exercises/figures/forward-backward-algorithm.png) (page ). We wish to compute P(Xk|e1:t)P(Xk|e1:t) for k=1,…,tk=1,…,t. This will be done with a divide-and-conquer approach.  
1. Suppose, for simplicity, that tt is odd, and let the halfway point be h=(t+1)/2h=(t+1)/2. Show that P(Xk|e1:t)P(Xk|e1:t) can be computed for k=1,…,hk=1,…,h given just the initial forward message f1:0f1:0, the backward message bh+1:tbh+1:t, and the evidence e1:he1:h.  
2. Show a similar result for the second half of the sequence.  
3. Given the results of (a) and (b), a recursive divide-and-conquer algorithm can be constructed by first running forward along the sequence and then backward from the end, storing just the required messages at the middle and the ends. Then the algorithm is called on each half. Write out the algorithm in detail.  
4. Compute the time and space complexity of the algorithm as a function of tt, the length of the sequence. How does this change if we divide the input into more than two pieces?

[Exercise 4 (flawed-viterbi-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_4/)

On page , we outlined a flawed procedure for finding the most likely state sequence, given an observation sequence. The procedure involves finding the most likely state at each time step, using smoothing, and returning the sequence composed of these states. Show that, for some temporal probability models and observation sequences, this procedure returns an impossible state sequence (i.e., the posterior probability of the sequence is zero).

[Exercise 5 (hmm-likelihood-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_5/)

Equation () describes the filtering process for the matrix formulation of HMMs. Give a similar equation for the calculation of likelihoods, which was described generically in Equation ().

[Exercise 6](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_6/)

Consider the vacuum worlds of Figure [4.18](https://aimacode.github.io/aima-exercises/figures/vacuum-maze-ch4-figure.png) (perfect sensing) and Figure [15.7](https://aimacode.github.io/aima-exercises/figures/vacuum-maze-hmm2-figure.png) (noisy sensing). Suppose that the robot receives an observation sequence such that, with perfect sensing, there is exactly one possible location it could be in. Is this location necessarily the most probable location under noisy sensing for sufficiently small noise probability ϵϵ? Prove your claim or find a counterexample.

[Exercise 7 (hmm-robust-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_7/)

In Section , the prior distribution over locations is uniform and the transition model assumes an equal probability of moving to any neighboring square. What if those assumptions are wrong? Suppose that the initial location is actually chosen uniformly from the northwest quadrant of the room and the action actually tends to move southeast. Keeping the HMM model fixed, explore the effect on localization and path accuracy as the southeasterly tendency increases, for different values of ϵϵ.

[Exercise 8 (roomba-viterbi-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_8/)

Consider a version of the vacuum robot (page ) that has the policy of going straight for as long as it can; only when it encounters an obstacle does it change to a new (randomly selected) heading. To model this robot, each state in the model consists of a *(location, heading)* pair. Implement this model and see how well the Viterbi algorithm can track a robot with this model. The robot’s policy is more constrained than the random-walk robot; does that mean that predictions of the most likely path are more accurate?

[Exercise 9](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_9/)

We have described three policies for the vacuum robot: (1) a uniform random walk, (2) a bias for wandering southeast, as described in Exercise [15.7](https://aimacode.github.io/aima-exercises/dbn-exercises/undefined), and (3) the policy described in Exercise [roomba-viterbi-exercise](https://aimacode.github.io/aima-exercises/dbn-exercises/). Suppose an observer is given the observation sequence from a vacuum robot, but is not sure which of the three policies the robot is following. What approach should the observer use to find the most likely path, given the observations? Implement the approach and test it. How much does the localization accuracy suffer, compared to the case in which the observer knows which policy the robot is following?

[Exercise 10](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_10/)

This exercise is concerned with filtering in an environment with no landmarks. Consider a vacuum robot in an empty room, represented by an n×mn×m rectangular grid. The robot’s location is hidden; the only evidence available to the observer is a noisy location sensor that gives an approximation to the robot’s location. If the robot is at location (x,y)(x,y) then with probability .1 the sensor gives the correct location, with probability .05 each it reports one of the 8 locations immediately surrounding (x,y)(x,y), with probability .025 each it reports one of the 16 locations that surround those 8, and with the remaining probability of .1 it reports “no reading.” The robot’s policy is to pick a direction and follow it with probability .8 on each step; the robot switches to a randomly selected new heading with probability .2 (or with probability 1 if it encounters a wall). Implement this as an HMM and do filtering to track the robot. How accurately can we track the robot’s path?

[Exercise 11](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_11/)

This exercise is concerned with filtering in an environment with no landmarks. Consider a vacuum robot in an empty room, represented by an n×mn×m rectangular grid. The robot’s location is hidden; the only evidence available to the observer is a noisy location sensor that gives an approximation to the robot’s location. If the robot is at location (x,y)(x,y) then with probability .1 the sensor gives the correct location, with probability .05 each it reports one of the 8 locations immediately surrounding (x,y)(x,y), with probability .025 each it reports one of the 16 locations that surround those 8, and with the remaining probability of .1 it reports “no reading.” The robot’s policy is to pick a direction and follow it with probability .7 on each step; the robot switches to a randomly selected new heading with probability .3 (or with probability 1 if it encounters a wall). Implement this as an HMM and do filtering to track the robot. How accurately can we track the robot’s path?

**A Bayesian network representation of a switching Kalman filter. The switching variable**StSt**is a discrete state variable whose value determines the transition model for the continuous state variables**XtXt**. For any discrete state**ii**, the transition model**P(Xt+1|Xt,St=i)P(Xt+1|Xt,St=i)**is a linear Gaussian model, just as in a regular Kalman filter. The transition model for the discrete state,**P(St+1|St)P(St+1|St)**, can be thought of as a matrix, as in a hidden Markov model.**

[Exercise 12 (switching-kf-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_12/)

Often, we wish to monitor a continuous-state system whose behavior switches unpredictably among a set of kk distinct “modes.” For example, an aircraft trying to evade a missile can execute a series of distinct maneuvers that the missile may attempt to track. A Bayesian network representation of such a **switching Kalman filter** model is shown in Figure [switching-kf-figure](https://aimacode.github.io/aima-exercises/dbn-exercises/#switching-kf-figure).  
  
1. Suppose that the discrete state StSt has kk possible values and that the prior continuous state estimate P(X0)P(X0) is a multivariate Gaussian distribution. Show that the prediction P(X1)P(X1) is a **mixture of Gaussians**—that is, a weighted sum of Gaussians such that the weights sum to 1.  
  
2. Show that if the current continuous state estimate P(Xt|e1:t)P(Xt|e1:t) is a mixture of mm Gaussians, then in the general case the updated state estimate P(Xt+1|e1:t+1)P(Xt+1|e1:t+1) will be a mixture of kmkm Gaussians.  
  
3. What aspect of the temporal process do the weights in the Gaussian mixture represent?  
  
The results in (a) and (b) show that the representation of the posterior grows without limit even for switching Kalman filters, which are among the simplest hybrid dynamic models.

[Exercise 13 (kalman-update-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_13/)

Complete the missing step in the derivation of Equation () on page , the first update step for the one-dimensional Kalman filter.

[Exercise 14 (kalman-variance-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_14/)

Let us examine the behavior of the variance update in Equation () (page ).  
1. Plot the value of σ2tσt2 as a function of tt, given various values for σ2xσx2 and σ2zσz2.  
2. Show that the update has a fixed point σ2σ2 such that σ2t→σ2σt2→σ2 as t→∞t→∞, and calculate the value of σ2σ2.  
3. Give a qualitative explanation for what happens as σ2x→0σx2→0 and as σ2z→0σz2→0.

[Exercise 15 (sleep1-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_15/)

A professor wants to know if students are getting enough sleep. Each day, the professor observes whether the students sleep in class, and whether they have red eyes. The professor has the following domain theory:  
- The prior probability of getting enough sleep, with no observations, is 0.7.  
- The probability of getting enough sleep on night tt is 0.8 given that the student got enough sleep the previous night, and 0.3 if not.  
- The probability of having red eyes is 0.2 if the student got enough sleep, and 0.7 if not.  
- The probability of sleeping in class is 0.1 if the student got enough sleep, and 0.3 if not.  
Formulate this information as a dynamic Bayesian network that the professor could use to filter or predict from a sequence of observations. Then reformulate it as a hidden Markov model that has only a single observation variable. Give the complete probability tables for the model.

[Exercise 16](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_16/)

A professor wants to know if students are getting enough sleep. Each day, the professor observes whether the students sleep in class, and whether they have red eyes. The professor has the following domain theory:  
- The prior probability of getting enough sleep, with no observations, is 0.7.  
- The probability of getting enough sleep on night tt is 0.8 given that the student got enough sleep the previous night, and 0.3 if not.  
- The probability of having red eyes is 0.2 if the student got enough sleep, and 0.7 if not.  
- The probability of sleeping in class is 0.1 if the student got enough sleep, and 0.3 if not.  
Formulate this information as a dynamic Bayesian network that the professor could use to filter or predict from a sequence of observations. Then reformulate it as a hidden Markov model that has only a single observation variable. Give the complete probability tables for the model.

[Exercise 17](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_17/)

For the DBN specified in Exercise [15.15](https://aimacode.github.io/aima-exercises/dbn-exercises/undefined) and for the evidence values  
e1=not red eyes, not sleeping in classe1=not red eyes, not sleeping in class  
e2=red eyes, not sleeping in classe2=red eyes, not sleeping in class  
e3=red eyes, sleeping in classe3=red eyes, sleeping in class  
perform the following computations:  
1. State estimation: Compute P(EnoughSleept|e1:t)P(EnoughSleept|e1:t) for each of t=1,2,3t=1,2,3.  
2. Smoothing: Compute P(EnoughSleept|e1:3)P(EnoughSleept|e1:3) for each of t=1,2,3t=1,2,3.  
3. Compare the filtered and smoothed probabilities for t=1t=1 and t=2t=2.

[Exercise 18](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_18/)

Suppose that a particular student shows up with red eyes and sleeps in class every day. Given the model described in Exercise [15.15](https://aimacode.github.io/aima-exercises/dbn-exercises/undefined), explain why the probability that the student had enough sleep the previous night converges to a fixed point rather than continuing to go down as we gather more days of evidence. What is the fixed point? Answer this both numerically (by computation) and analytically.

[Exercise 19 (battery-sequence-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_19/)

This exercise analyzes in more detail the persistent-failure model for the battery sensor in Figure [15.15](https://aimacode.github.io/aima-exercises/figures/battery-persistence-figure.png)(a) (page ).  
1. Figure [15.15](https://aimacode.github.io/aima-exercises/figures/battery-persistence-figure.png)(b) stops at t=32t=32. Describe qualitatively what should happen as t→∞t→∞ if the sensor continues to read 0.  
2. Suppose that the external temperature affects the battery sensor in such a way that transient failures become more likely as temperature increases. Show how to augment the DBN structure in Figure [15.15](https://aimacode.github.io/aima-exercises/figures/battery-persistence-figure.png)(a), and explain any required changes to the CPTs.  
3. Given the new network structure, can battery readings be used by the robot to infer the current temperature?

[Exercise 20 (dbn-elimination-exercise)](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_20/)

Consider applying the variable elimination algorithm to the umbrella DBN unrolled for three slices, where the query is P(R3|u1,u2,u3)P(R3|u1,u2,u3). Show that the space complexity of the algorithm—the size of the largest factor—is the same, regardless of whether the rain variables are eliminated in forward or backward order.

[Artificial Intelligence](https://aimacode.github.io/aima-exercises/) AIMA Exercises

窗体顶端

窗体底端

16. Making Simple Decisions

[Exercise 1 (almanac-game)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_1/)

(Adapted from David Heckerman.) This exercise concerns the **Almanac Game**, which is used by decision analysts to calibrate numeric estimation. For each of the questions that follow, give your best guess of the answer, that is, a number that you think is as likely to be too high as it is to be too low. Also give your guess at a 25th percentile estimate, that is, a number that you think has a 25% chance of being too high, and a 75% chance of being too low. Do the same for the 75th percentile. (Thus, you should give three estimates in all—low, median, and high—for each question.)  
1. Number of passengers who flew between New York and Los Angeles in 1989.  
2. Population of Warsaw in 1992.  
3. Year in which Coronado discovered the Mississippi River.  
4. Number of votes received by Jimmy Carter in the 1976 presidential election.  
5. Age of the oldest living tree, as of 2002.  
6. Height of the Hoover Dam in feet.  
7. Number of eggs produced in Oregon in 1985.  
8. Number of Buddhists in the world in 1992.  
9. Number of deaths due to AIDS in the United States in 1981.  
10. Number of U.S. patents granted in 1901.  
The correct answers appear after the last exercise of this chapter. From the point of view of decision analysis, the interesting thing is not how close your median guesses came to the real answers, but rather how often the real answer came within your 25% and 75% bounds. If it was about half the time, then your bounds are accurate. But if you’re like most people, you will be more sure of yourself than you should be, and fewer than half the answers will fall within the bounds. With practice, you can calibrate yourself to give realistic bounds, and thus be more useful in supplying information for decision making. Try this second set of questions and see if there is any improvement:  
1. Year of birth of Zsa Zsa Gabor.  
2. Maximum distance from Mars to the sun in miles.  
3. Value in dollars of exports of wheat from the United States in 1992.  
4. Tons handled by the port of Honolulu in 1991.  
5. Annual salary in dollars of the governor of California in 1993.  
6. Population of San Diego in 1990.  
7. Year in which Roger Williams founded Providence, Rhode Island.  
8. Height of Mt. Kilimanjaro in feet.  
9. Length of the Brooklyn Bridge in feet.  
10. Number of deaths due to automobile accidents in the United States in 1992.

[Exercise 2](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_2/)

Chris considers four used cars before buying the one with maximum expected utility. Pat considers ten cars and does the same. All other things being equal, which one is more likely to have the better car? Which is more likely to be disappointed with their car’s quality? By how much (in terms of standard deviations of expected quality)?

[Exercise 3](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_3/)

Chris considers five used cars before buying the one with maximum expected utility. Pat considers eleven cars and does the same. All other things being equal, which one is more likely to have the better car? Which is more likely to be disappointed with their car’s quality? By how much (in terms of standard deviations of expected quality)?

[Exercise 4 (St-Petersburg-exercise)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_4/)

In 1713, Nicolas Bernoulli stated a puzzle, now called the St. Petersburg paradox, which works as follows. You have the opportunity to play a game in which a fair coin is tossed repeatedly until it comes up heads. If the first heads appears on the nnth toss, you win 2n2n dollars.  
1. Show that the expected monetary value of this game is infinite.  
2. How much would you, personally, pay to play the game?  
3. Nicolas’s cousin Daniel Bernoulli resolved the apparent paradox in 1738 by suggesting that the utility of money is measured on a logarithmic scale (i.e., U(Sn)=alog2n+bU(Sn)=alog2⁡n+b, where SnSn is the state of having nn). What is the expected utility of the game under this assumption?  
4. What is the maximum amount that it would be rational to pay to play the game, assuming that one’s initial wealth is kk?

[Exercise 5](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_5/)

Write a computer program to automate the process in Exercise [assessment-exercise](https://aimacode.github.io/aima-exercises/decision-theory-exercises/). Try your program out on several people of different net worth and political outlook. Comment on the consistency of your results, both for an individual and across individuals.

[Exercise 6 (surprise-candy-exercise)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_6/)

The Surprise Candy Company makes candy in two flavors: 75% are strawberry flavor and 25% are anchovy flavor. Each new piece of candy starts out with a round shape; as it moves along the production line, a machine randomly selects a certain percentage to be trimmed into a square; then, each piece is wrapped in a wrapper whose color is chosen randomly to be red or brown. 70% of the strawberry candies are round and 70% have a red wrapper, while 90% of the anchovy candies are square and 90% have a brown wrapper. All candies are sold individually in sealed, identical, black boxes.  
Now you, the customer, have just bought a Surprise candy at the store but have not yet opened the box. Consider the three Bayes nets in Figure [3candy-figure](https://aimacode.github.io/aima-exercises/decision-theory-exercises/#3candy-figure).  
1. Which network(s) can correctly represent P(Flavor,Wrapper,Shape)P(Flavor,Wrapper,Shape)?  
2. Which network is the best representation for this problem?  
3. Does network (i) assert that P(Wrapper|Shape)P(Wrapper)P(Wrapper|Shape)P(Wrapper)?  
4. What is the probability that your candy has a red wrapper?  
5. In the box is a round candy with a red wrapper. What is the probability that its flavor is strawberry?  
6. A unwrapped strawberry candy is worth ss on the open market and an unwrapped anchovy candy is worth aa. Write an expression for the value of an unopened candy box.  
7. A new law prohibits trading of unwrapped candies, but it is still legal to trade wrapped candies (out of the box). Is an unopened candy box now worth more than less than, or the same as before?

**Three proposed Bayes nets for the Surprise Candy problem**

[Exercise 7 (surprise-candy-exercise)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_7/)

The Surprise Candy Company makes candy in two flavors: 70% are strawberry flavor and 30% are anchovy flavor. Each new piece of candy starts out with a round shape; as it moves along the production line, a machine randomly selects a certain percentage to be trimmed into a square; then, each piece is wrapped in a wrapper whose color is chosen randomly to be red or brown. 80% of the strawberry candies are round and 80% have a red wrapper, while 90% of the anchovy candies are square and 90% have a brown wrapper. All candies are sold individually in sealed, identical, black boxes.  
Now you, the customer, have just bought a Surprise candy at the store but have not yet opened the box. Consider the three Bayes nets in Figure [3candy-figure](https://aimacode.github.io/aima-exercises/decision-theory-exercises/#3candy-figure).  
1. Which network(s) can correctly represent P(Flavor,Wrapper,Shape)P(Flavor,Wrapper,Shape)?  
2. Which network is the best representation for this problem?  
3. Does network (i) assert that P(Wrapper|Shape)P(Wrapper)P(Wrapper|Shape)P(Wrapper)?  
4. What is the probability that your candy has a red wrapper?  
5. In the box is a round candy with a red wrapper. What is the probability that its flavor is strawberry?  
6. A unwrapped strawberry candy is worth ss on the open market and an unwrapped anchovy candy is worth aa. Write an expression for the value of an unopened candy box.  
7. A new law prohibits trading of unwrapped candies, but it is still legal to trade wrapped candies (out of the box). Is an unopened candy box now worth more than less than, or the same as before?

[Exercise 8](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_8/)

Prove that the judgments B≻AB≻A and C≻DC≻D in the Allais paradox (page ) violate the axiom of substitutability.

[Exercise 9](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_9/)

Consider the Allais paradox described on page : an agent who prefers BB over AA (taking the sure thing), and CC over DD (taking the higher EMV) is not acting rationally, according to utility theory. Do you think this indicates a problem for the agent, a problem for the theory, or no problem at all? Explain.

[Exercise 10](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_10/)

Tickets to a lottery cost 1. There are two possible prizes: a 10 payoff with probability 1/50, and a 1,000,000 payoff with probability 1/2,000,000. What is the expected monetary value of a lottery ticket? When (if ever) is it rational to buy a ticket? Be precise—show an equation involving utilities. You may assume current wealth of kk and that U(Sk)=0U(Sk)=0. You may also assume that U(Sk+10)=10×U(Sk+1)U(Sk+10)=10×U(Sk+1), but you may not make any assumptions about U(Sk+1,000,000)U(Sk+1,000,000). Sociological studies show that people with lower income buy a disproportionate number of lottery tickets. Do you think this is because they are worse decision makers or because they have a different utility function? Consider the value of contemplating the possibility of winning the lottery versus the value of contemplating becoming an action hero while watching an adventure movie.

[Exercise 11 (assessment-exercise)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_11/)

Assess your own utility for different incremental amounts of money by running a series of preference tests between some definite amount M1M1 and a lottery [p,M2;(1−p),0][p,M2;(1−p),0]. Choose different values of M1M1 and M2M2, and vary pp until you are indifferent between the two choices. Plot the resulting utility function.

[Exercise 12](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_12/)

How much is a micromort worth to you? Devise a protocol to determine this. Ask questions based both on paying to avoid risk and being paid to accept risk.

[Exercise 13 (kmax-exercise)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_13/)

Let continuous variables X1,…,XkX1,…,Xk be independently distributed according to the same probability density function f(x)f(x). Prove that the density function for max{X1,…,Xk}max{X1,…,Xk} is given by kf(x)(F(x))k−1kf(x)(F(x))k−1, where FF is the cumulative distribution for ff.

[Exercise 14](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_14/)

Economists often make use of an exponential utility function for money: U(x)=−e−x/RU(x)=−e−x/R, where RR is a positive constant representing an individual’s risk tolerance. Risk tolerance reflects how likely an individual is to accept a lottery with a particular expected monetary value (EMV) versus some certain payoff. As RR (which is measured in the same units as xx) becomes larger, the individual becomes less risk-averse.  
1. Assume Mary has an exponential utility function with R=$500R=$500. Mary is given the choice between receiving $500$500 with certainty (probability 1) or participating in a lottery which has a 60% probability of winning $5000 and a 40% probability of winning nothing. Assuming Marry acts rationally, which option would she choose? Show how you derived your answer.  
2. Consider the choice between receiving $100$100 with certainty (probability 1) or participating in a lottery which has a 50% probability of winning $500$500 and a 50% probability of winning nothing. Approximate the value of R (to 3 significant digits) in an exponential utility function that would cause an individual to be indifferent to these two alternatives. (You might find it helpful to write a short program to help you solve this problem.)

[Exercise 15](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_15/)

Economists often make use of an exponential utility function for money: U(x)=−e−x/RU(x)=−e−x/R, where RR is a positive constant representing an individual’s risk tolerance. Risk tolerance reflects how likely an individual is to accept a lottery with a particular expected monetary value (EMV) versus some certain payoff. As RR (which is measured in the same units as xx) becomes larger, the individual becomes less risk-averse.  
1. Assume Mary has an exponential utility function with R=$400R=$400. Mary is given the choice between receiving $400$400 with certainty (probability 1) or participating in a lottery which has a 60% probability of winning $5000 and a 40% probability of winning nothing. Assuming Marry acts rationally, which option would she choose? Show how you derived your answer.  
2. Consider the choice between receiving $100$100 with certainty (probability 1) or participating in a lottery which has a 50% probability of winning $500 and a 50% probability of winning nothing. Approximate the value of R (to 3 significant digits) in an exponential utility function that would cause an individual to be indifferent to these two alternatives. (You might find it helpful to write a short program to help you solve this problem.)

[Exercise 16](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_16/)

Alex is given the choice between two games. In Game 1, a fair coin is flipped and if it comes up heads, Alex receives $100$100. If the coin comes up tails, Alex receives nothing. In Game 2, a fair coin is flipped twice. Each time the coin comes up heads, Alex receives $50$50, and Alex receives nothing for each coin flip that comes up tails. Assuming that Alex has a monotonically increasing utility function for money in the range \[$0, $100\], show mathematically that if Alex prefers Game 2 to Game 1, then Alex is risk averse (at least with respect to this range of monetary amounts).  
Show that if X1X1 and X2X2 are preferentially independent of X3X3, and X2X2 and X3X3 are preferentially independent of X1X1, then X3X3 and X1X1 are preferentially independent of X2X2.

[Exercise 17 (airport-au-id-exercise)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_17/)

Repeat Exercise [16.21](https://aimacode.github.io/aima-exercises/decision-theory-exercises/undefined), using the action-utility representation shown in Figure [16.7](https://aimacode.github.io/aima-exercises/figures/airport-au-id-figure.png).

[Exercise 18](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_18/)

For either of the airport-siting diagrams from Exercises [16.21](https://aimacode.github.io/aima-exercises/decision-theory-exercises/undefined) and [16.17](https://aimacode.github.io/aima-exercises/decision-theory-exercises/undefined), to which conditional probability table entry is the utility most sensitive, given the available evidence?

[Exercise 19](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_19/)

Modify and extend the Bayesian network code in the code repository to provide for creation and evaluation of decision networks and the calculation of information value.

[Exercise 20](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_20/)

Consider a student who has the choice to buy or not buy a textbook for a course. We’ll model this as a decision problem with one Boolean decision node, BB, indicating whether the agent chooses to buy the book, and two Boolean chance nodes, MM, indicating whether the student has mastered the material in the book, and PP, indicating whether the student passes the course. Of course, there is also a utility node, UU. A certain student, Sam, has an additive utility function: 0 for not buying the book and -$100 for buying it; and $2000 for passing the course and 0 for not passing. Sam’s conditional probability estimates are as follows:

P(p|b,m)=0.9P(p|b,¬m)=0.5P(p|¬b,m)=0.8P(p|¬b,¬m)=0.3P(m|b)=0.9P(m|¬b)=0.7P(p|b,m)=0.9P(m|b)=0.9P(p|b,¬m)=0.5P(m|¬b)=0.7P(p|¬b,m)=0.8P(p|¬b,¬m)=0.3

You might think that PP would be independent of BB given MM, But this course has an open-book final—so having the book helps.  
1. Draw the decision network for this problem.  
2. Compute the expected utility of buying the book and of not buying it.  
3. What should Sam do?

[Exercise 21 (airport-id-exercise)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_21/)

This exercise completes the analysis of the airport-siting problem in Figure [16.6](https://aimacode.github.io/aima-exercises/figures/airport-id-figure.png)  
. 1. Provide reasonable variable domains, probabilities, and utilities for the network, assuming that there are three possible sites.  
2. Solve the decision problem.  
3. What happens if changes in technology mean that each aircraft generates half the noise?  
4. What if noise avoidance becomes three times more important?  
5. Calculate the VPI for AirTrafficAirTraffic, LitigationLitigation, and ConstructionConstruction in your model.

[Exercise 22 (car-vpi-exercise)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_22/)

(Adapted from Pearl [[Pearl:1988](https://aimacode.github.io/aima-exercises/decision-theory-exercises/)].) A used-car buyer can decide to carry out various tests with various costs (e.g., kick the tires, take the car to a qualified mechanic) and then, depending on the outcome of the tests, decide which car to buy. We will assume that the buyer is deciding whether to buy car c1c1, that there is time to carry out at most one test, and that t1t1 is the test of c1c1 and costs $50.  
A car can be in good shape (quality q+q+) or bad shape (quality q−q−), and the tests might help indicate what shape the car is in. Car c1c1 costs $1,500, and its market value is $2,000$2,000 if it is in good shape; if not, $700$700 in repairs will be needed to make it in good shape. The buyer’s estimate is that c1c1 has a 70% chance of being in good shape.  
1. Draw the decision network that represents this problem.  
2. Calculate the expected net gain from buying c1c1, given no test.  
3. Tests can be described by the probability that the car will pass or fail the test given that the car is in good or bad shape. We have the following information:  
P(pass(c1,t1)|q+(c1))=0.8P(pass(c1,t1)|q+(c1))=0.8  
P(pass(c1,t1)|q−(c1))=0.35P(pass(c1,t1)|q−(c1))=0.35  
Use Bayes’ theorem to calculate the probability that the car will pass (or fail) its test and hence the probability that it is in good (or bad) shape given each possible test outcome.  
4. Calculate the optimal decisions given either a pass or a fail, and their expected utilities.  
5. Calculate the value of information of the test, and derive an optimal conditional plan for the buyer.

[Exercise 23 (nonnegative-VPI-exercise)](https://aimacode.github.io/aima-exercises/decision-theory-exercises/ex_23/)

Recall the definition of *value of information* in Section .  
1. Prove that the value of information is nonnegative and order independent.  
2. Explain why it is that some people would prefer not to get some information—for example, not wanting to know the sex of their baby when an ultrasound is done.  
3. A function ff on sets is **submodular** if, for any element xx and any sets AA and BB such that A⊆BA⊆B, adding xx to AA gives a greater increase in ff than adding xx to BB:

A⊆B⇒(f(A∪{x})−f(A))≥(f(B∪{x})−f(B)) .A⊆B⇒(f(A∪{x})−f(A))≥(f(B∪{x})−f(B)) .

Submodularity captures the intuitive notion of *diminishing returns*. Is the value of information, viewed as a function ff on sets of possible observations, submodular? Prove this or find a counterexample.

17. Making Complex Decisions

[Exercise 1 (mdp-model-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_1/)

For the 4×34×3 world shown in Figure [15.4](https://aimacode.github.io/aima-exercises/figures/sequential-decision-world-figure.png)., calculate which squares can be reached from (1,1) by the action sequence [Up,Up,Right,Right,Right][Up,Up,Right,Right,Right] and with what probabilities. Explain how this computation is related to the prediction task (see Section [general-filtering-section](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/) for a hidden Markov model.

[Exercise 2 (mdp-model-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_2/)

For the 4×34×3 world shown in Figure [15.4](https://aimacode.github.io/aima-exercises/figures/sequential-decision-world-figure.png), calculate which squares can be reached from (1,1) by the action sequence [Right,Right,Right,Up,Up][Right,Right,Right,Up,Up] and with what probabilities. Explain how this computation is related to the prediction task (see Section ) for a hidden Markov model.

[Exercise 3](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_3/)

Select a specific member of the set of policies that are optimal for R(s)>0R(s)>0 as shown in Figure [17.2](https://aimacode.github.io/aima-exercises/figures/sequential-decision-policies-figure.png)(b), and calculate the fraction of time the agent spends in each state, in the limit, if the policy is executed forever. (*Hint*: Construct the state-to-state transition probability matrix corresponding to the policy and see Exercise [15.2](https://aimacode.github.io/aima-exercises/dbn-exercises/ex_2).)

[Exercise 4 (nonseparable-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_4/)

Suppose that we define the utility of a state sequence to be the *maximum* reward obtained in any state in the sequence. Show that this utility function does not result in stationary preferences between state sequences. Is it still possible to define a utility function on states such that MEU decision making gives optimal behavior?

[Exercise 5](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_5/)

Can any finite search problem be translated exactly into a Markov decision problem such that an optimal solution of the latter is also an optimal solution of the former? If so, explain *precisely* how to translate the problem and how to translate the solution back; if not, explain *precisely* why not (i.e., give a counterexample).

[Exercise 6 (reward-equivalence-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_6/)

Sometimes MDPs are formulated with a reward function R(s,a)R(s,a) that depends on the action taken or with a reward function R(s,a,s′)R(s,a,s′) that also depends on the outcome state.  
1. Write the Bellman equations for these formulations.  
2. Show how an MDP with reward function R(s,a,s′)R(s,a,s′) can be transformed into a different MDP with reward function R(s,a)R(s,a), such that optimal policies in the new MDP correspond exactly to optimal policies in the original MDP.  
3. Now do the same to convert MDPs with R(s,a)R(s,a) into MDPs with R(s)R(s).

[Exercise 7 (threshold-cost-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_7/)

For the environment shown in Figure [15.4](https://aimacode.github.io/aima-exercises/figures/sequential-decision-world-figure.png), find all the threshold values for R(s)R(s) such that the optimal policy changes when the threshold is crossed. You will need a way to calculate the optimal policy and its value for fixed R(s)R(s). (*Hint*: Prove that the value of any fixed policy varies linearly with R(s)R(s).)

[Exercise 8 (vi-contraction-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_8/)

Equation () on page  states that the Bellman operator is a contraction.  
1. Show that, for any functions ff and gg,

|maxaf(a)−maxag(a)|≤maxa|f(a)−g(a)| .|maxaf(a)−maxag(a)|≤maxa|f(a)−g(a)| .

2. Write out an expression for

|(BUi−BU′i)(s)||(BUi−BUi′)(s)|

and then apply the result from (1) to complete the proof that the Bellman operator is a contraction.

[Exercise 9](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_9/)

This exercise considers two-player MDPs that correspond to zero-sum, turn-taking games like those in Chapter [Adversarial Search](https://aimacode.github.io/aima-exercises/game-playing-exercises/). Let the players be AA and BB, and let R(s)R(s) be the reward for player AA in state ss. (The reward for BB is always equal and opposite.)  
1. Let UA(s)UA(s) be the utility of state ss when it is AA’s turn to move in ss, and let UB(s)UB(s) be the utility of state ss when it is BB’s turn to move in ss. All rewards and utilities are calculated from AA’s point of view (just as in a minimax game tree). Write down Bellman equations defining UA(s)UA(s) and UB(s)UB(s).  
2. Explain how to do two-player value iteration with these equations, and define a suitable termination criterion.  
3. Consider the game described in Figure [5.17](https://aimacode.github.io/aima-exercises/figures/line-game4-figure.png) on page . Draw the state space (rather than the game tree), showing the moves by AA as solid lines and moves by BB as dashed lines. Mark each state with R(s)R(s). You will find it helpful to arrange the states (sA,sB)(sA,sB) on a two-dimensional grid, using sAsA and sBsB as “coordinates.”  
4. Now apply two-player value iteration to solve this game, and derive the optimal policy.

**(a)**3×33×3**world for Exercise**[**3x3-mdp-exercise**](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/)**. The reward for each state is indicated. The upper right square is a terminal state. (b)**101×3101×3**world for Exercise**[**101x3-mdp-exercise**](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/)**(omitting 93 identical columns in the middle). The start state has reward 0.**

[Exercise 10 (3x3-mdp-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_10/)

Consider the 3×33×3 world shown in Figure [grid-mdp-figure](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/#grid-mdp-figure)(a). The transition model is the same as in the 4×34×3 Figure [15.4](https://aimacode.github.io/aima-exercises/figures/sequential-decision-world-figure.png): 80% of the time the agent goes in the direction it selects; the rest of the time it moves at right angles to the intended direction.  
Implement value iteration for this world for each value of rr below. Use discounted rewards with a discount factor of 0.99. Show the policy obtained in each case. Explain intuitively why the value of rr leads to each policy.  
1. r=−100r=−100  
2. r=−3r=−3  
3. r=0r=0  
4. r=+3r=+3

[Exercise 11 (101x3-mdp-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_11/)

Consider the 101×3101×3 world shown in Figure [grid-mdp-figure](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/#grid-mdp-figure)(b). In the start state the agent has a choice of two deterministic actions, *Up* or *Down*, but in the other states the agent has one deterministic action, *Right*. Assuming a discounted reward function, for what values of the discount γγ should the agent choose *Up* and for which *Down*? Compute the utility of each action as a function of γγ. (Note that this simple example actually reflects many real-world situations in which one must weigh the value of an immediate action versus the potential continual long-term consequences, such as choosing to dump pollutants into a lake.)

[Exercise 12](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_12/)

Consider an undiscounted MDP having three states, (1, 2, 3), with rewards −1−1, −2−2, 00, respectively. State 3 is a terminal state. In states 1 and 2 there are two possible actions: aa and bb. The transition model is as follows:  
- In state 1, action aa moves the agent to state 2 with probability 0.8 and makes the agent stay put with probability 0.2.  
- In state 2, action aa moves the agent to state 1 with probability 0.8 and makes the agent stay put with probability 0.2.  
- In either state 1 or state 2, action bb moves the agent to state 3 with probability 0.1 and makes the agent stay put with probability 0.9.  
Answer the following questions:  
1. What can be determined *qualitatively* about the optimal policy in states 1 and 2?  
2. Apply policy iteration, showing each step in full, to determine the optimal policy and the values of states 1 and 2. Assume that the initial policy has action bb in both states.  
3. What happens to policy iteration if the initial policy has action aa in both states? Does discounting help? Does the optimal policy depend on the discount factor?

[Exercise 13](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_13/)

Consider the 4×34×3 world shown in Figure [15.4](https://aimacode.github.io/aima-exercises/figures/sequential-decision-world-figure.png)  
. 1. Implement an environment simulator for this environment, such that the specific geography of the environment is easily altered. Some code for doing this is already in the online code repository.  
2. Create an agent that uses policy iteration, and measure its performance in the environment simulator from various starting states. Perform several experiments from each starting state, and compare the average total reward received per run with the utility of the state, as determined by your algorithm.  
3. Experiment with increasing the size of the environment. How does the run time for policy iteration vary with the size of the environment?

[Exercise 14 (policy-loss-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_14/)

How can the value determination algorithm be used to calculate the expected loss experienced by an agent using a given set of utility estimates UU and an estimated model PP, compared with an agent using correct values?

[Exercise 15 (4x3-pomdp-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_15/)

Let the initial belief state b0b0 for the 4×34×3 POMDP on page  be the uniform distribution over the nonterminal states, i.e., <19,19,19,19,19,19,19,19,19,0,0><19,19,19,19,19,19,19,19,19,0,0>. Calculate the exact belief state b1b1 after the agent moves and its sensor reports 1 adjacent wall. Also calculate b2b2 assuming that the same thing happens again.

[Exercise 16](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_16/)

What is the time complexity of dd steps of POMDP value iteration for a sensorless environment?

[Exercise 17 (2state-pomdp-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_17/)

Consider a version of the two-state POMDP on page  in which the sensor is 90% reliable in state 0 but provides no information in state 1 (that is, it reports 0 or 1 with equal probability). Analyze, either qualitatively or quantitatively, the utility function and the optimal policy for this problem.

[Exercise 18 (dominant-equilibrium-exercise)](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_18/)

Show that a dominant strategy equilibrium is a Nash equilibrium, but not vice versa.

[Exercise 19](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_19/)

In the children’s game of rock–paper–scissors each player reveals at the same time a choice of rock, paper, or scissors. Paper wraps rock, rock blunts scissors, and scissors cut paper. In the extended version rock–paper–scissors–fire–water, fire beats rock, paper, and scissors; rock, paper, and scissors beat water; and water beats fire. Write out the payoff matrix and find a mixed-strategy solution to this game.

[Exercise 20](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_20/)

Solve the game of *three*-finger Morra.

[Exercise 21](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_21/)

In the *Prisoner’s Dilemma*, consider the case where after each round, Alice and Bob have probability XX meeting again. Suppose both players choose the perpetual punishment strategy (where each will choose refuserefuse unless the other player has ever played testifytestify). Assume neither player has played testifytestify thus far. What is the expected future total payoff for choosing to testifytestify versus refuserefuse when X=.2X=.2? How about when X=.05X=.05? For what value of XX is the expected future total payoff the same whether one chooses to testifytestify or refuserefuse in the current round?

[Exercise 22](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_22/)

The following payoff matrix, from @Blinder:1983 by way of [Bernstein:1996](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/), shows a game between politicians and the Federal Reserve.

Pol:contractPol:donothingPol:expandFed:contractF=7,P=1F=8,P=2F=3,P=3Fed:donothingF=9,P=4F=5,P=5F=2,P=7Fed:expandF=6,P=6F=4,P=9F=1,P=8Fed:contractFed:donothingFed:expandPol:contractF=7,P=1F=9,P=4F=6,P=6Pol:donothingF=8,P=2F=5,P=5F=4,P=9Pol:expandF=3,P=3F=2,P=7F=1,P=8

Politicians can expand or contract fiscal policy, while the Fed can expand or contract monetary policy. (And of course either side can choose to do nothing.) Each side also has preferences for who should do what—neither side wants to look like the bad guys. The payoffs shown are simply the rank orderings: 9 for first choice through 1 for last choice. Find the Nash equilibrium of the game in pure strategies. Is this a Pareto-optimal solution? You might wish to analyze the policies of recent administrations in this light.

[Exercise 23](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_23/)

A Dutch auction is similar in an English auction, but rather than starting the bidding at a low price and increasing, in a Dutch auction the seller starts at a high price and gradually lowers the price until some buyer is willing to accept that price. (If multiple bidders accept the price, one is arbitrarily chosen as the winner.) More formally, the seller begins with a price pp and gradually lowers pp by increments of dd until at least one buyer accepts the price. Assuming all bidders act rationally, is it true that for arbitrarily small dd, a Dutch auction will always result in the bidder with the highest value for the item obtaining the item? If so, show mathematically why. If not, explain how it may be possible for the bidder with highest value for the item not to obtain it.

[Exercise 24](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_24/)

Imagine an auction mechanism that is just like an ascending-bid auction, except that at the end, the winning bidder, the one who bid bmaxbmax, pays only bmax/2bmax/2 rather than bmaxbmax. Assuming all agents are rational, what is the expected revenue to the auctioneer for this mechanism, compared with a standard ascending-bid auction?

[Exercise 25](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/ex_25/)

Teams in the National Hockey League historically received 2 points for winning a game and 0 for losing. If the game is tied, an overtime period is played; if nobody wins in overtime, the game is a tie and each team gets 1 point. But league officials felt that teams were playing too conservatively in overtime (to avoid a loss), and it would be more exciting if overtime produced a winner. So in 1999 the officials experimented in mechanism design: the rules were changed, giving a team that loses in overtime 1 point, not 0. It is still 2 points for a win and 1 for a tie.  
1. Was hockey a zero-sum game before the rule change? After?  
2. Suppose that at a certain time tt in a game, the home team has probability pp of winning in regulation time, probability 0.78−p0.78−p of losing, and probability 0.22 of going into overtime, where they have probability qq of winning, .9−q.9−q of losing, and .1 of tying. Give equations for the expected value for the home and visiting teams.  
3. Imagine that it were legal and ethical for the two teams to enter into a pact where they agree that they will skate to a tie in regulation time, and then both try in earnest to win in overtime. Under what conditions, in terms of pp and qq, would it be rational for both teams to agree to this pact?  
4. [Longley+Sankaran:2005](https://aimacode.github.io/aima-exercises/complex-decisions-exercises/) report that since the rule change, the percentage of games with a winner in overtime went up 18.2%, as desired, but the percentage of overtime games also went up 3.6%. What does that suggest about possible collusion or conservative play after the rule change?

18. Learning from Examples

[Exercise 1 (infant-language-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_1/)

Consider the problem faced by an infant learning to speak and understand a language. Explain how this process fits into the general learning model. Describe the percepts and actions of the infant, and the types of learning the infant must do. Describe the subfunctions the infant is trying to learn in terms of inputs and outputs, and available example data.

[Exercise 2](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_2/)

Repeat Exercise [18.1](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_1) for the case of learning to play tennis (or some other sport with which you are familiar). Is this supervised learning or reinforcement learning?

[Exercise 3](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_3/)

Draw a decision tree for the problem of deciding whether to move forward at a road intersection, given that the light has just turned green.

[Exercise 4](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_4/)

We never test the same attribute twice along one path in a decision tree. Why not?

[Exercise 5](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_5/)

Suppose we generate a training set from a decision tree and then apply decision-tree learning to that training set. Is it the case that the learning algorithm will eventually return the correct tree as the training-set size goes to infinity? Why or why not?

[Exercise 6 (leaf-classification-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_6/)

In the recursive construction of decision trees, it sometimes happens that a mixed set of positive and negative examples remains at a leaf node, even after all the attributes have been used. Suppose that we have pp positive examples and nn negative examples.  
1. Show that the solution used by DECISION-TREE-LEARNING, which picks the majority classification, minimizes the absolute error over the set of examples at the leaf.  
2. Show that the **class probability** p/(p+n)p/(p+n) minimizes the sum of squared errors.

[Exercise 7 (nonnegative-gain-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_7/)

Suppose that an attribute splits the set of examples EE into subsets EkEk and that each subset has pkpk positive examples and nknk negative examples. Show that the attribute has strictly positive information gain unless the ratio pk/(pk+nk)pk/(pk+nk) is the same for all kk.

[Exercise 8](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_8/)

Consider the following data set comprised of three binary input attributes (A1,A2A1,A2, and A3A3) and one binary output:

Examplex1x2x3x4x5A111011A200111A301010Output y00011ExampleA1A2A3Output yx11000x21010x30100x41111x51101

Use the algorithm in Figure [18.5](https://aimacode.github.io/aima-exercises/figures/DTL-algorithm.png) (page ) to learn a decision tree for these data. Show the computations made to determine the attribute to split at each node.

[Exercise 9](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_9/)

Construct a data set (set of examples with attributes and classifications) that would cause the decision-tree learning algorithm to find a non-minimal-sized tree. Show the tree constructed by the algorithm and the minimal-sized tree that you can generate by hand.

[Exercise 10](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_10/)

A decision *graph* is a generalization of a decision tree that allows nodes (i.e., attributes used for splits) to have multiple parents, rather than just a single parent. The resulting graph must still be acyclic. Now, consider the XOR function of *three* binary input attributes, which produces the value 1 if and only if an odd number of the three input attributes has value 1.  
1. Draw a minimal-sized decision *tree* for the three-input XOR function.  
2. Draw a minimal-sized decision *graph* for the three-input XOR function.

[Exercise 11 (pruning-DTL-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_11/)

This exercise considers χ2χ2 pruning of decision trees (Section   
. 1. Create a data set with two input attributes, such that the information gain at the root of the tree for both attributes is zero, but there is a decision tree of depth 2 that is consistent with all the data. What would χ2χ2 pruning do on this data set if applied bottom up? If applied top down?  
2. Modify DECISION-TREE-LEARNING to include χ2χ2-pruning. You might wish to consult Quinlan [[Quinlan:1986](https://aimacode.github.io/aima-exercises/concept-learning-exercises/)] or [[Kearns+Mansour:1998](https://aimacode.github.io/aima-exercises/concept-learning-exercises/)] for details.

[Exercise 12 (missing-value-DTL-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_12/)

The standard DECISION-TREE-LEARNING algorithm described in the chapter does not handle cases in which some examples have missing attribute values.  
1. First, we need to find a way to classify such examples, given a decision tree that includes tests on the attributes for which values can be missing. Suppose that an example xx has a missing value for attribute AA and that the decision tree tests for AA at a node that xx reaches. One way to handle this case is to pretend that the example has *all* possible values for the attribute, but to weight each value according to its frequency among all of the examples that reach that node in the decision tree. The classification algorithm should follow all branches at any node for which a value is missing and should multiply the weights along each path. Write a modified classification algorithm for decision trees that has this behavior.  
2. Now modify the information-gain calculation so that in any given collection of examples CC at a given node in the tree during the construction process, the examples with missing values for any of the remaining attributes are given “as-if” values according to the frequencies of those values in the set CC.

[Exercise 13 (gain-ratio-DTL-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_13/)

In Section , we noted that attributes with many different possible values can cause problems with the gain measure. Such attributes tend to split the examples into numerous small classes or even singleton classes, thereby appearing to be highly relevant according to the gain measure. The **gain-ratio** criterion selects attributes according to the ratio between their gain and their intrinsic information content—that is, the amount of information contained in the answer to the question, “What is the value of this attribute?” The gain-ratio criterion therefore tries to measure how efficiently an attribute provides information on the correct classification of an example. Write a mathematical expression for the information content of an attribute, and implement the gain ratio criterion in DECISION-TREE-LEARNING.

[Exercise 14](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_14/)

Suppose you are running a learning experiment on a new algorithm for Boolean classification. You have a data set consisting of 100 positive and 100 negative examples. You plan to use leave-one-out cross-validation and compare your algorithm to a baseline function, a simple majority classifier. (A majority classifier is given a set of training data and then always outputs the class that is in the majority in the training set, regardless of the input.) You expect the majority classifier to score about 50% on leave-one-out cross-validation, but to your surprise, it scores zero every time. Can you explain why?

[Exercise 15](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_15/)

Suppose that a learning algorithm is trying to find a consistent hypothesis when the classifications of examples are actually random. There are nn Boolean attributes, and examples are drawn uniformly from the set of 2n2n possible examples. Calculate the number of examples required before the probability of finding a contradiction in the data reaches 0.5.

[Exercise 16](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_16/)

Construct a *decision list* to classify the data below. Select tests to be as small as possible (in terms of attributes), breaking ties among tests with the same number of attributes by selecting the one that classifies the greatest number of examples correctly. If multiple tests have the same number of attributes and classify the same number of examples, then break the tie using attributes with lower index numbers (e.g., select A1A1 over A2A2).

Examplex1x2x3x4x5x6x7x8A111001000A200111100A301010011A401001110y11101010ExampleA1A2A3A4yx110001x210111x301001x401100x511011x601010x700111x800100

[Exercise 17](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_17/)

Prove that a decision list can represent the same function as a decision tree while using at most as many rules as there are leaves in the decision tree for that function. Give an example of a function represented by a decision list using strictly fewer rules than the number of leaves in a minimal-sized decision tree for that same function.

[Exercise 18 (DL-expressivity-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_18/)

This exercise concerns the expressiveness of decision lists (Section ).  
1. Show that decision lists can represent any Boolean function, if the size of the tests is not limited.  
2. Show that if the tests can contain at most kk literals each, then decision lists can represent any function that can be represented by a decision tree of depth kk.

[Exercise 19 (knn-mean-mode)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_19/)

Suppose a 77-nearest-neighbors regression search returns {7,6,8,4,7,11,100}{7,6,8,4,7,11,100} as the 7 nearest yy values for a given xx value. What is the value of y^y^ that minimizes the L1L1 loss function on this data? There is a common name in statistics for this value as a function of the yy values; what is it? Answer the same two questions for the L2L2 loss function.

[Exercise 20 (knn-mean-mode)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_20/)

Suppose a 77-nearest-neighbors regression search returns {4,2,8,4,9,11,100}{4,2,8,4,9,11,100} as the 7 nearest yy values for a given xx value. What is the value of y^y^ that minimizes the L1L1 loss function on this data? There is a common name in statistics for this value as a function of the yy values; what is it? Answer the same two questions for the L2L2 loss function.

[Exercise 21 (svm-ellipse-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_21/)

Figure [kernel-machine-figure](https://aimacode.github.io/aima-exercises/concept-learning-exercises/) showed how a circle at the origin can be linearly separated by mapping from the features (x1,x2)(x1,x2) to the two dimensions (x21,x22)(x12,x22). But what if the circle is not located at the origin? What if it is an ellipse, not a circle? The general equation for a circle (and hence the decision boundary) is (x1−a)2+(x2−b)2−r20(x1−a)2+(x2−b)2−r20, and the general equation for an ellipse is c(x1−a)2+d(x2−b)2−10c(x1−a)2+d(x2−b)2−10.  
1. Expand out the equation for the circle and show what the weights wiwi would be for the decision boundary in the four-dimensional feature space (x1,x2,x21,x22)(x1,x2,x12,x22). Explain why this means that any circle is linearly separable in this space.  
2. Do the same for ellipses in the five-dimensional feature space (x1,x2,x21,x22,x1x2)(x1,x2,x12,x22,x1x2).

[Exercise 22 (svm-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_22/)

Construct a support vector machine that computes the xor function. Use values of +1 and –1 (instead of 1 and 0) for both inputs and outputs, so that an example looks like ([−1,1],1)([−1,1],1) or ([−1,−1],−1)([−1,−1],−1). Map the input [x1,x2][x1,x2] into a space consisting of x1x1 and x1x2x1x2. Draw the four input points in this space, and the maximal margin separator. What is the margin? Now draw the separating line back in the original Euclidean input space.

[Exercise 23 (ensemble-error-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_23/)

Consider an ensemble learning algorithm that uses simple majority voting among KK learned hypotheses. Suppose that each hypothesis has error ϵϵ and that the errors made by each hypothesis are independent of the others’. Calculate a formula for the error of the ensemble algorithm in terms of KK and ϵϵ, and evaluate it for the cases where K=5K=5, 10, and 20 and ϵ=0.1ϵ=0.1, 0.2, and 0.4. If the independence assumption is removed, is it possible for the ensemble error to be *worse* than ϵϵ?

[Exercise 24](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_24/)

Construct by hand a neural network that computes the xor function of two inputs. Make sure to specify what sort of units you are using.

[Exercise 25](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_25/)

A simple perceptron cannot represent xor (or, generally, the parity function of its inputs). Describe what happens to the weights of a four-input, hard-threshold perceptron, beginning with all weights set to 0.1, as examples of the parity function arrive.

[Exercise 26 (linear-separability-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_26/)

Recall from Chapter [Learning From Examples](https://aimacode.github.io/aima-exercises/concept-learning-exercises/) that there are 22n22n distinct Boolean functions of nn inputs. How many of these are representable by a threshold perceptron?

[Exercise 27](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_27/)

Consider the following set of examples, each with six inputs and one target output:

Examplex1x2x3x4x5x6TA11010001A21011001A31010101A41100111A51111001A61000111A71000100A80111011A90110110A100001100A110101010A120001010A130110110A140111000ExampleA1A2A3A4A5A6A7A8A9A10A11A12A13A14x111111110000000x200011001101011x311101001100011x401001001011101x500110110110010x600010101101110T11111101000000

1. Run the perceptron learning rule on these data and show the final weights.  
2. Run the decision tree learning rule, and show the resulting decision tree.  
3. Comment on your results.

[Exercise 28 (perceptron-ML-gradient-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_28/)

Section  (page ) noted that the output of the logistic function could be interpreted as a *probability* pp assigned by the model to the proposition that f(x)1f(x)1; the probability that f(x)0f(x)0 is therefore 1−p1−p. Write down the probability pp as a function of xx and calculate the derivative of logplog⁡p with respect to each weight wiwi. Repeat the process for log(1−p)log⁡(1−p). These calculations give a learning rule for minimizing the negative-log-likelihood loss function for a probabilistic hypothesis. Comment on any resemblance to other learning rules in the chapter.

[Exercise 29 (linear-nn-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_29/)

Suppose you had a neural network with linear activation functions. That is, for each unit the output is some constant cc times the weighted sum of the inputs.  
1. Assume that the network has one hidden layer. For a given assignment to the weights ww, write down equations for the value of the units in the output layer as a function of ww and the input layer xx, without any explicit mention of the output of the hidden layer. Show that there is a network with no hidden units that computes the same function.  
2. Repeat the calculation in part (a), but this time do it for a network with any number of hidden layers.  
3. Suppose a network with one hidden layer and linear activation functions has nn input and output nodes and hh hidden nodes. What effect does the transformation in part (a) to a network with no hidden layers have on the total number of weights? Discuss in particular the case h≪nh≪n.

[Exercise 30](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_30/)

Implement a data structure for layered, feed-forward neural networks, remembering to provide the information needed for both forward evaluation and backward propagation. Using this data structure, write a function NEURAL-NETWORK-OUTPUT that takes an example and a network and computes the appropriate output values.

[Exercise 31](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_31/)

Suppose that a training set contains only a single example, repeated 100 times. In 80 of the 100 cases, the single output value is 1; in the other 20, it is 0. What will a back-propagation network predict for this example, assuming that it has been trained and reaches a global optimum? (*Hint:* to find the global optimum, differentiate the error function and set it to zero.)

[Exercise 32](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_32/)

The neural network whose learning performance is measured in Figure [18.25](https://aimacode.github.io/aima-exercises/figures/restaurant-back-prop-figure.png) has four hidden nodes. This number was chosen somewhat arbitrarily. Use a cross-validation method to find the best number of hidden nodes.

[Exercise 33 (embedding-separability-exercise)](https://aimacode.github.io/aima-exercises/concept-learning-exercises/ex_33/)

Consider the problem of separating NN data points into positive and negative examples using a linear separator. Clearly, this can always be done for N2N2 points on a line of dimension d1d1, regardless of how the points are labeled or where they are located (unless the points are in the same place).  
1. Show that it can always be done for N3N3 points on a plane of dimension d2d2, unless they are collinear.  
2. Show that it cannot always be done for N4N4 points on a plane of dimension d2d2.  
3. Show that it can always be done for N4N4 points in a space of dimension d3d3, unless they are coplanar.  
4. Show that it cannot always be done for N5N5 points in a space of dimension d3d3.  
5. The ambitious student may wish to prove that NN points in general position (but not N+1N+1) are linearly separable in a space of dimension N−1N−1.

19. Knowledge in Learning

[Exercise 1 (dbsig-exercise)](https://aimacode.github.io/aima-exercises/ilp-exercises/ex_1/)

Show, by translating into conjunctive normal form and applying resolution, that the conclusion drawn on page  concerning Brazilians is sound.

[Exercise 2](https://aimacode.github.io/aima-exercises/ilp-exercises/ex_2/)

For each of the following determinations, write down the logical representation and explain why the determination is true (if it is):  
1. Design and denomination determine the mass of a coin.  
2. For a given program, input determines output.  
3. Climate, food intake, exercise, and metabolism determine weight gain and loss.  
4. Baldness is determined by the baldness (or lack thereof) of one’s maternal grandfather.

[Exercise 3](https://aimacode.github.io/aima-exercises/ilp-exercises/ex_3/)

For each of the following determinations, write down the logical representation and explain why the determination is true (if it is):  
1. Zip code determines the state (U.S.).  
2. Design and denomination determine the mass of a coin.  
3. Climate, food intake, exercise, and metabolism determine weight gain and loss.  
4. Baldness is determined by the baldness (or lack thereof) of one’s maternal grandfather.

[Exercise 4](https://aimacode.github.io/aima-exercises/ilp-exercises/ex_4/)

Would a probabilistic version of determinations be useful? Suggest a definition.

[Exercise 5 (ir-step-exercise)](https://aimacode.github.io/aima-exercises/ilp-exercises/ex_5/)

Fill in the missing values for the clauses C1C1 or C2C2 (or both) in the following sets of clauses, given that CC is the resolvent of C1C1 and C2C2:  
1. C=True⇒P(A,B)C=True⇒P(A,B), C1=P(x,y)⇒Q(x,y)C1=P(x,y)⇒Q(x,y), C2=??C2=??.  
2. C=True⇒P(A,B)C=True⇒P(A,B), C1=??C1=??, C2=??C2=??.  
3. C=P(x,y)⇒P(x,f(y))C=P(x,y)⇒P(x,f(y)), C1=??C1=??, C2=??C2=??.  
If there is more than one possible solution, provide one example of each different kind.

[Exercise 6 (prolog-ir-exercise)](https://aimacode.github.io/aima-exercises/ilp-exercises/ex_6/)

Suppose one writes a logic program that carries out a resolution inference step. That is, let Resolve(c1,c2,c)Resolve(c1,c2,c) succeed if cc is the result of resolving c1c1 and c2c2. Normally, ResolveResolve would be used as part of a theorem prover by calling it with c1c1 and c2c2 instantiated to particular clauses, thereby generating the resolvent cc. Now suppose instead that we call it with cc instantiated and c1c1 and c2c2 uninstantiated. Will this succeed in generating the appropriate results of an inverse resolution step? Would you need any special modifications to the logic programming system for this to work?

[Exercise 7 (foil-literals-exercise)](https://aimacode.github.io/aima-exercises/ilp-exercises/ex_7/)

Suppose that is considering adding a literal to a clause using a binary predicate PP and that previous literals (including the head of the clause) contain five different variables.  
1. How many functionally different literals can be generated? Two literals are functionally identical if they differ only in the names of the \*new\* variables that they contain.  
2. Can you find a general formula for the number of different literals with a predicate of arity rr when there are nn variables previously used?  
3. Why does not allow literals that contain no previously used variables?

[Exercise 8](https://aimacode.github.io/aima-exercises/ilp-exercises/ex_8/)

Using the data from the family tree in Figure [19.11](https://aimacode.github.io/aima-exercises/figures/family2-figure.png), or a subset thereof, apply the algorithm to learn a definition for the AncestorAncestor predicate.

20. Learning Probabilistic Models

[Exercise 1 (bayes-candy-exercise)](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_1/)

The data used for Figure [20.1](https://aimacode.github.io/aima-exercises/figures/bayes-candy-figure.png) on page  can be viewed as being generated by h5h5. For each of the other four hypotheses, generate a data set of length 100 and plot the corresponding graphs for P(hi|d1,…,dN)P(hi|d1,…,dN) and P(DN+1=lime|d1,…,dN)P(DN+1=lime|d1,…,dN). Comment on your results.

[Exercise 2](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_2/)

Repeat Exercise [20.1](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_1), this time plotting the values of P(DN+1=lime|hMAP)P(DN+1=lime|hMAP) and P(DN+1=lime|hML)P(DN+1=lime|hML).

[Exercise 3 (candy-trade-exercise)](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_3/)

Suppose that Ann’s utilities for cherry and lime candies are cAcA and ℓAℓA, whereas Bob’s utilities are cBcB and ℓBℓB. (But once Ann has unwrapped a piece of candy, Bob won’t buy it.) Presumably, if Bob likes lime candies much more than Ann, it would be wise for Ann to sell her bag of candies once she is sufficiently sure of its lime content. On the other hand, if Ann unwraps too many candies in the process, the bag will be worth less. Discuss the problem of determining the optimal point at which to sell the bag. Determine the expected utility of the optimal procedure, given the prior distribution from Section .

[Exercise 4](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_4/)

Two statisticians go to the doctor and are both given the same prognosis: A 40% chance that the problem is the deadly disease AA, and a 60% chance of the fatal disease BB. Fortunately, there are anti-AA and anti-BB drugs that are inexpensive, 100% effective, and free of side-effects. The statisticians have the choice of taking one drug, both, or neither. What will the first statistician (an avid Bayesian) do? How about the second statistician, who always uses the maximum likelihood hypothesis?  
The doctor does some research and discovers that disease BB actually comes in two versions, dextro-BB and levo-BB, which are equally likely and equally treatable by the anti-BB drug. Now that there are three hypotheses, what will the two statisticians do?

[Exercise 5 (BNB-exercise)](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_5/)

Explain how to apply the boosting method of Chapter [Learning From Examples](https://aimacode.github.io/aima-exercises/concept-learning-exercises/) to naive Bayes learning. Test the performance of the resulting algorithm on the restaurant learning problem.

[Exercise 6 (linear-regression-exercise)](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_6/)

Consider NN data points (xj,yj)(xj,yj), where the yjyjs are generated from the xjxjs according to the linear Gaussian model in Equation (). Find the values of θ1θ1, θ2θ2, and σσ that maximize the conditional log likelihood of the data.

[Exercise 7 (noisy-OR-ML-exercise)](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_7/)

Consider the noisy-OR model for fever described in Section . Explain how to apply maximum-likelihood learning to fit the parameters of such a model to a set of complete data. (*Hint*: use the chain rule for partial derivatives.)

[Exercise 8 (beta-integration-exercise)](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_8/)

This exercise investigates properties of the Beta distribution defined in Equation ().  
1. By integrating over the range [0,1][0,1], show that the normalization constant for the distribution [a,b][a,b] is given by α=Γ(a+b)/Γ(a)Γ(b)α=Γ(a+b)/Γ(a)Γ(b) where Γ(x)Γ(x) is the **Gamma function**, defined by Γ(x+1)x⋅Γ(x)Γ(x+1)x⋅Γ(x) and Γ(1)1Γ(1)1. (For integer xx, Γ(x+1)x!Γ(x+1)x!.)  
2. Show that the mean is a/(a+b)a/(a+b).  
3. Find the mode(s) (the most likely value(s) of θθ).  
4. Describe the distribution [ϵ,ϵ][ϵ,ϵ] for very small ϵϵ. What happens as such a distribution is updated?

[Exercise 9 (ML-parents-exercise)](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_9/)

Consider an arbitrary Bayesian network, a complete data set for that network, and the likelihood for the data set according to the network. Give a simple proof that the likelihood of the data cannot decrease if we add a new link to the network and recompute the maximum-likelihood parameter values.

[Exercise 10](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_10/)

Consider a single Boolean random variable YY (the “classification”). Let the prior probability P(Y=true)P(Y=true) be ππ. Let’s try to find ππ, given a training set D=(y1,…,yN)D=(y1,…,yN) with NN independent samples of YY. Furthermore, suppose pp of the NN are positive and nn of the NN are negative.  
1. Write down an expression for the likelihood of DD (i.e., the probability of seeing this particular sequence of examples, given a fixed value of ππ) in terms of ππ, pp, and nn.  
2. By differentiating the log likelihood LL, find the value of ππ that maximizes the likelihood.  
3. Now suppose we add in kk Boolean random variables X1,X2,…,XkX1,X2,…,Xk (the “attributes”) that describe each sample, and suppose we assume that the attributes are conditionally independent of each other given the goal YY. Draw the Bayes net corresponding to this assumption.  
4. Write down the likelihood for the data including the attributes, using the following additional notation:  
- αiαi is P(Xi=true∥Y=true)P(Xi=true‖Y=true).  
- βiβi is P(Xi=true∥Y=false)P(Xi=true‖Y=false).  
- p+ipi+ is the count of samples for which Xi=trueXi=true and Y=trueY=true.  
- n+ini+ is the count of samples for which Xi=falseXi=false and Y=trueY=true.  
- p−ipi− is the count of samples for which Xi=trueXi=true and Y=falseY=false.  
- n−ini− is the count of samples for which Xi=falseXi=false and Y=falseY=false.  
\[*Hint*: consider first the probability of seeing a single example with specified values for X1,X2,…,XkX1,X2,…,Xk and YY.\]  
5. By differentiating the log likelihood LL, find the values of αiαi and βiβi (in terms of the various counts) that maximize the likelihood and say in words what these values represent.  
6. Let k=2k=2, and consider a data set with 4 all four possible examples of thexor function. Compute the maximum likelihood estimates of ππ, α1α1, α2α2, β1β1, and β2β2.  
7. Given these estimates of ππ, α1α1, α2α2, β1β1, and β2β2, what are the posterior probabilities P(Y=true|x1,x2)P(Y=true|x1,x2) for each example?

[Exercise 11](https://aimacode.github.io/aima-exercises/bayesian-learning-exercises/ex_11/)

Consider the application of EM to learn the parameters for the network in Figure [20.11](https://aimacode.github.io/aima-exercises/figures/mixture-networks-figure.png)(a), given the true parameters in Equation (). 1. Explain why the EM algorithm would not work if there were just two attributes in the model rather than three. 2. Show the calculations for the first iteration of EM starting from Equation (). 3. What happens if we start with all the parameters set to the same value pp? (*Hint*: you may find it helpful to investigate this empirically before deriving the general result.) 4. Write out an expression for the log likelihood of the tabulated candy data on page  in terms of the parameters, calculate the partial derivatives with respect to each parameter, and investigate the nature of the fixed point reached in part (c).

21. Reinforcement Learning

[Exercise 1](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_1/)

Implement a passive learning agent in a simple environment, such as the 4×34×3 world. For the case of an initially unknown environment model, compare the learning performance of the direct utility estimation, TD, and ADP algorithms. Do the comparison for the optimal policy and for several random policies. For which do the utility estimates converge faster? What happens when the size of the environment is increased? (Try environments with and without obstacles.)

[Exercise 2](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_2/)

Chapter [Making Complex Decisions](https://aimacode.github.io/aima-exercises/concept-decisions-exercise/) defined a **proper policy** for an MDP as one that is guaranteed to reach a terminal state. Show that it is possible for a passive ADP agent to learn a transition model for which its policy ππ is improper even if ππ is proper for the true MDP; with such models, the POLICY-EVALUATION step may fail if γ1γ1. Show that this problem cannot arise if POLICY-EVALUATION is applied to the learned model only at the end of a trial.

[Exercise 3 (prioritized-sweeping-exercise)](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_3/)

Starting with the passive ADP agent, modify it to use an approximate ADP algorithm as discussed in the text. Do this in two steps:  
1. Implement a priority queue for adjustments to the utility estimates. Whenever a state is adjusted, all of its predecessors also become candidates for adjustment and should be added to the queue. The queue is initialized with the state from which the most recent transition took place. Allow only a fixed number of adjustments.  
2. Experiment with various heuristics for ordering the priority queue, examining their effect on learning rates and computation time.

[Exercise 4](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_4/)

The direct utility estimation method in Section  uses distinguished terminal states to indicate the end of a trial. How could it be modified for environments with discounted rewards and no terminal states?

[Exercise 5](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_5/)

Write out the parameter update equations for TD learning with

U^(x,y)=θ0+θ1x+θ2y+θ3(x−xg)2+(y−yg)2−−−−−−−−−−−−−−−−−√ .U^(x,y)=θ0+θ1x+θ2y+θ3(x−xg)2+(y−yg)2 .

[Exercise 6](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_6/)

Adapt the vacuum world (Chapter [Intelligent Agents](https://aimacode.github.io/aima-exercises/agents-exercises/) for reinforcement learning by including rewards for squares being clean. Make the world observable by providing suitable percepts. Now experiment with different reinforcement learning agents. Is function approximation necessary for success? What sort of approximator works for this application?

[Exercise 7 (approx-LMS-exercise)](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_7/)

Implement an exploring reinforcement learning agent that uses direct utility estimation. Make two versions—one with a tabular representation and one using the function approximator in Equation (). Compare their performance in three environments:  
1. The 4×34×3 world described in the chapter.  
2. A 10×1010×10 world with no obstacles and a +1 reward at (10,10).  
3. A 10×1010×10 world with no obstacles and a +1 reward at (5,5).

[Exercise 8](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_8/)

Devise suitable features for reinforcement learning in stochastic grid worlds (generalizations of the 4×34×3 world) that contain multiple obstacles and multiple terminal states with rewards of +1+1 or −1−1.

[Exercise 9](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_9/)

Extend the standard game-playing environment (Chapter [Adversarial Search](https://aimacode.github.io/aima-exercises/game-playing-exercises/)) to incorporate a reward signal. Put two reinforcement learning agents into the environment (they may, of course, share the agent program) and have them play against each other. Apply the generalized TD update rule (Equation ()) to update the evaluation function. You might wish to start with a simple linear weighted evaluation function and a simple game, such as tic-tac-toe.

[Exercise 10 (10x10-exercise)](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_10/)

Compute the true utility function and the best linear approximation in xx and yy (as in Equation ()) for the following environments:  
1. A 10×1010×10 world with a single +1+1 terminal state at (10,10).  
2. As in (a), but add a −1−1 terminal state at (10,1).  
3. As in (b), but add obstacles in 10 randomly selected squares.  
4. As in (b), but place a wall stretching from (5,2) to (5,9).  
5. As in (a), but with the terminal state at (5,5).  
The actions are deterministic moves in the four directions. In each case, compare the results using three-dimensional plots. For each environment, propose additional features (besides xx and yy) that would improve the approximation and show the results.

[Exercise 11](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_11/)

Implement the REINFORCE and PEGASUS algorithms and apply them to the 4×34×3 world, using a policy family of your own choosing. Comment on the results.

[Exercise 12](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_12/)

Investigate the application of reinforcement learning ideas to the modeling of human and animal behavior.

[Exercise 13](https://aimacode.github.io/aima-exercises/reinforcement-learning-exercises/ex_13/)

Is reinforcement learning an appropriate abstract model for evolution? What connection exists, if any, between hardwired reward signals and evolutionary fitness?

22. Natural Language Processing

[Exercise 1](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_1/)

This exercise explores the quality of the nn-gram model of language. Find or create a monolingual corpus of 100,000 words or more. Segment it into words, and compute the frequency of each word. How many distinct words are there? Also count frequencies of bigrams (two consecutive words) and trigrams (three consecutive words). Now use those frequencies to generate language: from the unigram, bigram, and trigram models, in turn, generate a 100-word text by making random choices according to the frequency counts. Compare the three generated texts with actual language. Finally, calculate the perplexity of each model.

[Exercise 2](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_2/)

Write a program to do **segmentation** of words without spaces. Given a string, such as the URL “thelongestlistofthelongeststuffatthelongestdomainnameatlonglast.com,” return a list of component words: [“the,” “longest,” “list,” ……]. This task is useful for parsing URLs, for spelling correction when words runtogether, and for languages such as Chinese that do not have spaces between words. It can be solved with a unigram or bigram word model and a dynamic programming algorithm similar to the Viterbi algorithm.

[Exercise 3](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_3/)

*Zipf’s law* of word distribution states the following: Take a large corpus of text, count the frequency of every word in the corpus, and then rank these frequencies in decreasing order. Let fIfI be the IIth largest frequency in this list; that is, f1f1 is the frequency of the most common word (usually “the”), f2f2 is the frequency of the second most common word, and so on. Zipf’s law states that fIfI is approximately equal to α/Iα/I for some constant αα. The law tends to be highly accurate except for very small and very large values of II.

[Exercise 4](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_4/)

Choose a corpus of at least 20,000 words of online text, and verify Zipf’s law experimentally. Define an error measure and find the value of αα where Zipf’s law best matches your experimental data. Create a log–log graph plotting fIfI vs. II and α/Iα/I vs. II. (On a log–log graph, the function α/Iα/I is a straight line.) In carrying out the experiment, be sure to eliminate any formatting tokens (e.g., HTML tags) and normalize upper and lower case.

[Exercise 5](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_5/)

(Adapted from [Jurafsky+Martin:2000](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/).) In this exercise you will develop a classifier for authorship: given a text, the classifier predicts which of two candidate authors wrote the text. Obtain samples of text from two different authors. Separate them into training and test sets. Now train a language model on the training set. You can choose what features to use; nn-grams of words or letters are the easiest, but you can add additional features that you think may help. Then compute the probability of the text under each language model and chose the most probable model. Assess the accuracy of this technique. How does accuracy change as you alter the set of features? This subfield of linguistics is called **stylometry**; its successes include the identification of the author of the disputed *Federalist Papers* [Mosteller+Wallace:1964](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/) and some disputed works of Shakespeare [Hope:1994](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/). [Khmelev+Tweedie:2001](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/) produce good results with a simple letter bigram model.

[Exercise 6](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_6/)

This exercise concerns the classification of spam email. Create a corpus of spam email and one of non-spam mail. Examine each corpus and decide what features appear to be useful for classification: unigram words? bigrams? message length, sender, time of arrival? Then train a classification algorithm (decision tree, naive Bayes, SVM, logistic regression, or some other algorithm of your choosing) on a training set and report its accuracy on a test set.

[Exercise 7](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_7/)

Create a test set of ten queries, and pose them to three major Web search engines. Evaluate each one for precision at 1, 3, and 10 documents. Can you explain the differences between engines?

[Exercise 8](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_8/)

Try to ascertain which of the search engines from the previous exercise are using case folding, stemming, synonyms, and spelling correction.

[Exercise 9](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_9/)

Estimate how much storage space is necessary for the index to a 100 billion-page corpus of Web pages. Show the assumptions you made.

[Exercise 10](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_10/)

Write a regular expression or a short program to extract company names. Test it on a corpus of business news articles. Report your recall and precision.

[Exercise 11](https://aimacode.github.io/aima-exercises/nlp-communicating-exercises/ex_11/)

Consider the problem of trying to evaluate the quality of an IR system that returns a ranked list of answers (like most Web search engines). The appropriate measure of quality depends on the presumed model of what the searcher is trying to achieve, and what strategy she employs. For each of the following models, propose a corresponding numeric measure.  
1. The searcher will look at the first twenty answers returned, with the objective of getting as much relevant information as possible.  
2. The searcher needs only one relevant document, and will go down the list until she finds the first one.  
3. The searcher has a fairly narrow query and is able to examine all the answers retrieved. She wants to be sure that she has seen everything in the document collection that is relevant to her query. (E.g., a lawyer wants to be sure that she has found *all* relevant precedents, and is willing to spend considerable resources on that.)  
4. The searcher needs just one document relevant to the query, and can afford to pay a research assistant for an hour’s work looking through the results. The assistant can look through 100 retrieved documents in an hour. The assistant will charge the searcher for the full hour regardless of whether he finds it immediately or at the end of the hour.  
5. The searcher will look through all the answers. Examining a document has cost $ A; finding a relevant document has value $ B; failing to find a relevant document has cost $ C for each relevant document not found.  
6. The searcher wants to collect as many relevant documents as possible, but needs steady encouragement. She looks through the documents in order. If the documents she has looked at so far are mostly good, she will continue; otherwise, she will stop.

23. Natural Language for Communication

[Exercise 1 (washing-clothes-exercise)](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_1/)

Read the following text once for understanding, and remember as much of it as you can. There will be a test later.  
> The procedure is actually quite simple. First you arrange things into different groups. Of course, one pile may be sufficient depending on how much there is to do. If you have to go somewhere else due to lack of facilities that is the next step, otherwise you are pretty well set. It is important not to overdo things. That is, it is better to do too few things at once than too many. In the short run this may not seem important but complications can easily arise. A mistake is expensive as well. At first the whole procedure will seem complicated. Soon, however, it will become just another facet of life. It is difficult to foresee any end to the necessity for this task in the immediate future, but then one can never tell. After the procedure is completed one arranges the material into different groups again. Then they can be put into their appropriate places. Eventually they will be used once more and the whole cycle will have to be repeated. However, this is part of life.

[Exercise 2](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_2/)

An *HMM grammar* is essentially a standard HMM whose state variable is NN (nonterminal, with values such as DetDet, AdjectiveAdjective, NounNoun and so on) and whose evidence variable is WW (word, with values such as isis, duckduck, and so on). The HMM model includes a prior P(N0)P(N0), a transition model P(Nt+1|Nt)P(Nt+1|Nt), and a sensor model P(Wt|Nt)P(Wt|Nt). Show that every HMM grammar can be written as a PCFG. [Hint: start by thinking about how the HMM prior can be represented by PCFG rules for the sentence symbol. You may find it helpful to illustrate for the particular HMM with values AA, BB for NN and values xx, yy for WW.]

[Exercise 3](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_3/)

Consider the following PCFG for simple verb phrases:  
> 0.1: VP →→ Verb  
> 0.2: VP →→ Copula Adjective  
> 0.5: VP →→ Verb the Noun  
> 0.2: VP →→ VP Adverb  
> 0.5: Verb →→ is  
> 0.5: Verb →→ shoots  
> 0.8: Copula →→ is  
> 0.2: Copula →→ seems  
> 0.5: Adjective →→ **unwell**  
> 0.5: Adjective →→ **well**  
> 0.5: Adverb →→ **well**  
> 0.5: Adverb →→ **badly**  
> 0.6: Noun →→ **duck**  
> 0.4: Noun →→ **well**  
1. Which of the following have a nonzero probability as a VP? (i) shoots the duck well well well(ii) seems the well well(iii) shoots the unwell well badly  
2. What is the probability of generating “is well well”?  
3. What types of ambiguity are exhibited by the phrase in (b)?  
4. Given any PCFG, is it possible to calculate the probability that the PCFG generates a string of exactly 10 words?

[Exercise 4](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_4/)

Consider the following simple PCFG for noun phrases:  
> 0.6: NP →→ Det\ AdjString\ Noun  
> 0.4: NP →→ Det\ NounNounCompound  
> 0.5: AdjString →→ Adj\ AdjString  
> 0.5: AdjString →→ ΛΛ  
> 1.0: NounNounCompound →→ Noun  
> 0.8: Det →→ **the**  
> 0.2: Det →→ **a**  
> 0.5: Adj →→ **small**  
> 0.5: Adj →→ **green**  
> 0.6: Noun →→ **village**  
> 0.4: Noun →→ **green**  
where ΛΛ denotes the empty string.  
1. What is the longest NP that can be generated by this grammar? (i) three words(ii) four words(iii) infinitely many words  
2. Which of the following have a nonzero probability of being generated as complete NPs? (i) a small green village(ii) a green green green(iii) a small village green  
3. What is the probability of generating “the green green”?  
4. What types of ambiguity are exhibited by the phrase in (c)?  
5. Given any PCFG and any finite word sequence, is it possible to calculate the probability that the sequence was generated by the PCFG?

[Exercise 5](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_5/)

Outline the major differences between Java (or any other computer language with which you are familiar) and English, commenting on the “understanding” problem in each case. Think about such things as grammar, syntax, semantics, pragmatics, compositionality, context-dependence, lexical ambiguity, syntactic ambiguity, reference finding (including pronouns), background knowledge, and what it means to “understand” in the first place.

[Exercise 6](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_6/)

This exercise concerns grammars for very simple languages.  
1. Write a context-free grammar for the language anbnanbn.  
2. Write a context-free grammar for the palindrome language: the set of all strings whose second half is the reverse of the first half.  
3. Write a context-sensitive grammar for the duplicate language: the set of all strings whose second half is the same as the first half.

[Exercise 7](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_7/)

Consider the sentence “Someone walked slowly to the supermarket” and a lexicon consisting of the following words:  
Pronoun→someoneVerb→walkedPronoun→someoneVerb→walked  
Adv→slowlyPrep→toAdv→slowlyPrep→to  
Article→theNoun→supermarketArticle→theNoun→supermarket  
Which of the following three grammars, combined with the lexicon, generates the given sentence? Show the corresponding parse tree(s).

(A):(B):(C):S→NP VPS→NP VPS→NP VPNP→PronounNP→PronounNP→PronounNP→Article NounNP→NounNP→Article NPVP→VP PPNP→Article NPVP→Verb AdvVP→VP Adv AdvVP→Verb VmodAdv→Adv AdvVP→VerbVmod→Adv VmodAdv→PPPP→Prep NPVmod→AdvPP→Prep NPNP→NounAdv→PPNP→NounPP→Prep NP(A):(B):(C):S→NP VPS→NP VPS→NP VPNP→PronounNP→PronounNP→PronounNP→Article NounNP→NounNP→Article NPVP→VP PPNP→Article NPVP→Verb AdvVP→VP Adv AdvVP→Verb VmodAdv→Adv AdvVP→VerbVmod→Adv VmodAdv→PPPP→Prep NPVmod→AdvPP→Prep NPNP→NounAdv→PPNP→NounPP→Prep NP

For each of the preceding three grammars, write down three sentences of English and three sentences of non-English generated by the grammar. Each sentence should be significantly different, should be at least six words long, and should include some new lexical entries (which you should define). Suggest ways to improve each grammar to avoid generating the non-English sentences.

[Exercise 8](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_8/)

Collect some examples of time expressions, such as “two o’clock,” “midnight,” and “12:46.” Also think up some examples that are ungrammatical, such as “thirteen o’clock” or “half past two fifteen.” Write a grammar for the time language.

[Exercise 9](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_9/)

Some linguists have argued as follows:  
  
Children learning a language hear only *positive examples* of the language and no *negative examples*. Therefore, the hypothesis that “every possible sentence is in the language” is consistent with all the observed examples. Moreover, this is the simplest consistent hypothesis. Furthermore, all grammars for languages that are supersets of the true language are also consistent with the observed data. Yet children do induce (more or less) the right grammar. It follows that they begin with very strong innate grammatical constraints that rule out all of these more general hypotheses *a priori*.  
Comment on the weak point(s) in this argument from a statistical learning viewpoint.

[Exercise 10 (chomsky-form-exercise)](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_10/)

In this exercise you will transform ε0ε0 into Chomsky Normal Form (CNF). There are five steps: (a) Add a new start symbol, (b) Eliminate ϵϵ rules, (c) Eliminate multiple words on right-hand sides, (d) Eliminate rules of the form (X→X→YY), (e) Convert long right-hand sides into binary rules.  
1. The start symbol, SS, can occur only on the left-hand side in CNF. Replace SS everywhere by a new symbol S′S′ and add a rule of the form SS →→S′S′.  
2. The empty string, ϵϵ cannot appear on the right-hand side in CNF. ε0ε0 does not have any rules with ϵϵ, so this is not an issue.  
3. A word can appear on the right-hand side in a rule only of the form (XX →→*word*). Replace each rule of the form (XX →→…*word* …) with (XX →→…W′W′ …) and (W′W′ →→*word*), using a new symbol W′W′.  
4. A rule (XX →→YY) is not allowed in CNF; it must be (XX →→YY ZZ) or (XX →→*word*). Replace each rule of the form (XX →→YY) with a set of rules of the form (XX →→…), one for each rule (YY →→…), where (…) indicates one or more symbols.  
5. Replace each rule of the form (XX →→YY ZZ …) with two rules, (XX →→YY Z′Z′) and (Z′Z′ →→ZZ …), where Z′Z′ is a new symbol.  
Show each step of the process and the final set of rules.

[Exercise 11](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_11/)

Consider the following toy grammar:  
> S→NP VPS→NP VP  
> NP→NounNP→Noun  
> NP→NP and NPNP→NP and NP  
> NP→NP PPNP→NP PP  
> VP→VerbVP→Verb  
> VP→VP and VPVP→VP and VP  
> VP→VP PPVP→VP PP  
> PP→Prep NPPP→Prep NP  
> Noun→Sally ;pools ;streams ;swimsNoun→Sally ;pools ;streams ;swims  
> Prep→inPrep→in  
> Verb→pools ;streams ;swimsVerb→pools ;streams ;swims  
1. Show all the parse trees in this grammar for the sentence “Sally swims in streams and pools.”  
2. Show all the table entries that would be made by a (non-probabalistic) CYK parser on this sentence.

[Exercise 12(exercise-subj-verb-agree)](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_12/)

Using DCG notation, write a grammar for a language that is just like ε1ε1, except that it enforces agreement between the subject and verb of a sentence and thus does not generate ungrammatical sentences such as “I smells the wumpus.”

[Exercise 13](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_13/)

Consider the following PCFG:  
> S→NP VP[1.0]S→NP VP[1.0]  
> NP→Noun[0.6] | Pronoun[0.4]NP→Noun[0.6] | Pronoun[0.4]  
> VP→Verb NP[0.8] | Modal Verb[0.2]VP→Verb NP[0.8] | Modal Verb[0.2]  
> Noun→can[0.1] | fish[0.3] | ...Noun→can[0.1] | fish[0.3] | ...  
> Pronoun→I[0.4] | ...Pronoun→I[0.4] | ...  
> Verb→can[0.01] | fish[0.1] | ...Verb→can[0.01] | fish[0.1] | ...  
> Modal→can[0.3] | ...Modal→can[0.3] | ...  
The sentence “I can fish” has two parse trees with this grammar. Show the two trees, their prior probabilities, and their conditional probabilities, given the sentence.

[Exercise 14](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_14/)

An augmented context-free grammar can represent languages that a regular context-free grammar cannot. Show an augmented context-free grammar for the language anbncnanbncn. The allowable values for augmentation variables are 1 and SUCCESSOR(n)SUCCESSOR(n), where nn is a value. The rule for a sentence in this language is

S(n)→A(n)B(n)C(n) .S(n)→A(n)B(n)C(n) .

Show the rule(s) for each of AA, BB, and CC.

[Exercise 15](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_15/)

Augment the ε1ε1 grammar so that it handles article–noun agreement. That is, make sure that “agents” and “an agent” are NPNPs, but “agent” and “an agents” are not.

[Exercise 16](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_16/)

Consider the following sentence (from *The New York Times,* July 28, 2008):  
> Banks struggling to recover from multibillion-dollar loans on real > estate are curtailing loans to American businesses, depriving even > healthy companies of money for expansion and hiring. 1. Which of the words in this sentence are lexically ambiguous?  
2. Find two cases of syntactic ambiguity in this sentence (there are more than two.)  
3. Give an instance of metaphor in this sentence.  
4. Can you find semantic ambiguity?

[Exercise 17 (washing-clothes2-exercise)](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_17/)

Without looking back at Exercise [23.1](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_1), answer the following questions:  
1. What are the four steps that are mentioned?  
2. What step is left out?  
3. What is “the material” that is mentioned in the text?  
4. What kind of mistake would be expensive?  
5. Is it better to do too few things or too many? Why?

[Exercise 18](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_18/)

Select five sentences and submit them to an online translation service. Translate them from English to another language and back to English. Rate the resulting sentences for grammaticality and preservation of meaning. Repeat the process; does the second round of iteration give worse results or the same results? Does the choice of intermediate language make a difference to the quality of the results? If you know a foreign language, look at the translation of one paragraph into that language. Count and describe the errors made, and conjecture why these errors were made.

[Exercise 19](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_19/)

The DiDi values for the sentence in Figure [23.13](https://aimacode.github.io/aima-exercises/figures/mt-alignment-figure.png) sum to 0. Will that be true of every translation pair? Prove it or give a counterexample.

[Exercise 20](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_20/)

(Adapted from [[Knight:1999](https://aimacode.github.io/aima-exercises/nlp-english-exercises/)].) Our translation model assumes that, after the phrase translation model selects phrases and the distortion model permutes them, the language model can unscramble the permutation. This exercise investigates how sensible that assumption is. Try to unscramble these proposed lists of phrases into the correct order:  
1. have, programming, a, seen, never, I, language, better  
2. loves, john, mary  
3. is the, communication, exchange of, intentional, information brought, by, about, the production, perception of, and signs, from, drawn, a, of, system, signs, conventional, shared  
4. created, that, we hold these, to be, all men, truths, are, equal, self-evident  
Which ones could you do? What type of knowledge did you draw upon? Train a bigram model from a training corpus, and use it to find the highest-probability permutation of some sentences from a test corpus. Report on the accuracy of this model.

[Exercise 21](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_21/)

Calculate the most probable path through the HMM in Figure [23.16](https://aimacode.github.io/aima-exercises/figures/sr-hmm-figure.png) for the output sequence [C1,C2,C3,C4,C4,C6,C7][C1,C2,C3,C4,C4,C6,C7]. Also give its probability.

[Exercise 22](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_22/)

We forgot to mention that the text in Exercise [23.1](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_1) is entitled “Washing Clothes.” Reread the text and answer the questions in Exercise [23.17](https://aimacode.github.io/aima-exercises/nlp-english-exercises/ex_17). Did you do better this time? Bransford and Johnson [[Bransford+Johnson:1973](https://aimacode.github.io/aima-exercises/nlp-english-exercises/)] used this text in a controlled experiment and found that the title helped significantly. What does this tell you about how language and memory works?

24. Perception

[Exercise 1](https://aimacode.github.io/aima-exercises/perception-exercises/ex_1/)

In the shadow of a tree with a dense, leafy canopy, one sees a number of light spots. Surprisingly, they all appear to be circular. Why? After all, the gaps between the leaves through which the sun shines are not likely to be circular.

[Exercise 2](https://aimacode.github.io/aima-exercises/perception-exercises/ex_2/)

Consider a picture of a white sphere floating in front of a black backdrop. The image curve separating white pixels from black pixels is sometimes called the “outline” of the sphere. Show that the outline of a sphere, viewed in a perspective camera, can be an ellipse. Why do spheres not look like ellipses to you?

[Exercise 3](https://aimacode.github.io/aima-exercises/perception-exercises/ex_3/)

Consider an infinitely long cylinder of radius rr oriented with its axis along the yy-axis. The cylinder has a Lambertian surface and is viewed by a camera along the positive zz-axis. What will you expect to see in the image if the cylinder is illuminated by a point source at infinity located on the positive xx-axis? Draw the contours of constant brightness in the projected image. Are the contours of equal brightness uniformly spaced?

[Exercise 4](https://aimacode.github.io/aima-exercises/perception-exercises/ex_4/)

Edges in an image can correspond to a variety of events in a scene. Consider Figure [24.4](https://aimacode.github.io/aima-exercises/figures/illuminationfigure.png) (page , and assume that it is a picture of a real three-dimensional scene. Identify ten different brightness edges in the image, and for each, state whether it corresponds to a discontinuity in (a) depth, (b) surface orientation, (c) reflectance, or (d) illumination.

[Exercise 5](https://aimacode.github.io/aima-exercises/perception-exercises/ex_5/)

A stereoscopic system is being contemplated for terrain mapping. It will consist of two CCD cameras, each having 512×512512×512 pixels on a 10 cm ×× 10 cm square sensor. The lenses to be used have a focal length of 16 cm, with the focus fixed at infinity. For corresponding points (u1,v1u1,v1) in the left image and (u2,v2u2,v2) in the right image, v1=v2v1=v2 because the xx-axes in the two image planes are parallel to the epipolar lines—the lines from the object to the camera. The optical axes of the two cameras are parallel. The baseline between the cameras is 1 meter.  
1. If the nearest distance to be measured is 16 meters, what is the largest disparity that will occur (in pixels)?  
2. What is the distance resolution at 16 meters, due to the pixel spacing?  
3. What distance corresponds to a disparity of one pixel?

[Exercise 6](https://aimacode.github.io/aima-exercises/perception-exercises/ex_6/)

Which of the following are true, and which are false?  
1. Finding corresponding points in stereo images is the easiest phase of the stereo depth-finding process.  
2. Shape-from-texture can be done by projecting a grid of light-stripes onto the scene.  
3. Lines with equal lengths in the scene always project to equal lengths in the image.  
4. Straight lines in the image necessarily correspond to straight lines in the scene.

[Exercise 7](https://aimacode.github.io/aima-exercises/perception-exercises/ex_7/)

Which of the following are true, and which are false?  
1. Finding corresponding points in stereo images is the easiest phase of the stereo depth-finding process.  
2. In stereo views of the same scene, greater accuracy is obtained in the depth calculations if the two camera positions are farther apart.  
3. Lines with equal lengths in the scene always project to equal lengths in the image.  
4. Straight lines in the image necessarily correspond to straight lines in the scene.

**Top view of a two-camera vision system observing a bottle with a wall behind it.**

[Exercise 8](https://aimacode.github.io/aima-exercises/perception-exercises/ex_8/)

(Courtesy of Pietro Perona.) Figure [bottle-figure](https://aimacode.github.io/aima-exercises/perception-exercises/#bottle-figure) shows two cameras at X and Y observing a scene. Draw the image seen at each camera, assuming that all named points are in the same horizontal plane. What can be concluded from these two images about the relative distances of points A, B, C, D, and E from the camera baseline, and on what basis?

25. Robotics

[Exercise 1 (mcl-biasdness-exercise)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_1/)

Monte Carlo localization is *biased* for any finite sample size—i.e., the expected value of the location computed by the algorithm differs from the true expected value—because of the way particle filtering works. In this question, you are asked to quantify this bias.  
To simplify, consider a world with four possible robot locations: X={x1,x2,x3,x4}X={x1,x2,x3,x4}. Initially, we draw N≥N≥ samples uniformly from among those locations. As usual, it is perfectly acceptable if more than one sample is generated for any of the locations XX. Let ZZ be a Boolean sensor variable characterized by the following conditional probabilities:

P(z|x1)=0.8P(z|x1)=0.2P(z|x2)=0.4P(z|x2)=0.6P(z|x3)=0.1P(z|x3)=0.9P(z|x4)=0.1P(z|x4)=0.9P(z|x1)=0.8P(z|x1)=0.2P(z|x2)=0.4P(z|x2)=0.6P(z|x3)=0.1P(z|x3)=0.9P(z|x4)=0.1P(z|x4)=0.9

MCL uses these probabilities to generate particle weights, which are subsequently normalized and used in the resampling process. For simplicity, let us assume we generate only one new sample in the resampling process, regardless of NN. This sample might correspond to any of the four locations in XX. Thus, the sampling process defines a probability distribution over XX.  
1. What is the resulting probability distribution over XX for this new sample? Answer this question separately for N=1,…,10N=1,…,10, and for N=∞N=∞.  
2. The difference between two probability distributions PP and QQ can be measured by the KL divergence, which is defined as

KL(P,Q)=∑iP(xi)logP(xi)Q(xi) .KL(P,Q)=∑iP(xi)log⁡P(xi)Q(xi) .

What are the KL divergences between the distributions in (a) and the true posterior?  
3. What modification of the problem formulation (not the algorithm!) would guarantee that the specific estimator above is unbiased even for finite values of NN? Provide at least two such modifications (each of which should be sufficient).

[Exercise 2 (mcl-implement-exercise)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_2/)

Implement Monte Carlo localization for a simulated robot with range sensors. A grid map and range data are available from the code repository at [aima.cs.berkeley.edu](http://aima.cs.berkeley.edu/). You should demonstrate successful global localization of the robot.

**A Robot manipulator in two of its possible configurations.**

[Exercise 3 (AB-manipulator-ex)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_3/)

Consider a robot with two simple manipulators, as shown in figure [figRobot2](https://aimacode.github.io/aima-exercises/robotics-exercises/#figRobot2). Manipulator A is a square block of side 2 which can slide back and on a rod that runs along the x-axis from x=−−10 to x=10. Manipulator B is a square block of side 2 which can slide back and on a rod that runs along the y-axis from y=-10 to y=10. The rods lie outside the plane of manipulation, so the rods do not interfere with the movement of the blocks. A configuration is then a pair ⟨x,y⟩⟨x,y⟩ where xx is the x-coordinate of the center of manipulator A and where yy is the y-coordinate of the center of manipulator B. Draw the configuration space for this robot, indicating the permitted and excluded zones.

[Exercise 4](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_4/)

Suppose that you are working with the robot in Exercise [25.3](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_3) and you are given the problem of finding a path from the starting configuration of figure [figRobot2](https://aimacode.github.io/aima-exercises/robotics-exercises/#figRobot2) to the ending configuration. Consider a potential function

D(A,Goal)2+D(B,Goal)2+1D(A,B)2D(A,Goal)2+D(B,Goal)2+1D(A,B)2

where D(A,B)D(A,B) is the distance between the closest points of A and B.  
1. Show that hill climbing in this potential field will get stuck in a local minimum.  
2. Describe a potential field where hill climbing will solve this particular problem. You need not work out the exact numerical coefficients needed, just the general form of the solution. (Hint: Add a term that “rewards" the hill climber for moving A out of B’s way, even in a case like this where this does not reduce the distance from A to B in the above sense.)

[Exercise 5 (inverse-kinematics-exercise)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_5/)

Consider the robot arm shown in Figure [25.14](https://aimacode.github.io/aima-exercises/figures/FigArm1.png). Assume that the robot’s base element is 60cm long and that its upper arm and forearm are each 40cm long. As argued on page , the inverse kinematics of a robot is often not unique. State an explicit closed-form solution of the inverse kinematics for this arm. Under what exact conditions is the solution unique?

[Exercise 6 (inverse-kinematics-exercise)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_6/)

Consider the robot arm shown in Figure [25.14](https://aimacode.github.io/aima-exercises/figures/FigArm1.png). Assume that the robot’s base element is 70cm long and that its upper arm and forearm are each 50cm long. As argued on page , the inverse kinematics of a robot is often not unique. State an explicit closed-form solution of the inverse kinematics for this arm. Under what exact conditions is the solution unique?

[Exercise 7 (voronoi-exercise)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_7/)

Implement an algorithm for calculating the Voronoi diagram of an arbitrary 2D environment, described by an n×nn×n Boolean array. Illustrate your algorithm by plotting the Voronoi diagram for 10 interesting maps. What is the complexity of your algorithm?

[Exercise 8 (confspace-exercise)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_8/)

This exercise explores the relationship between workspace and configuration space using the examples shown in Figure [FigEx2](https://aimacode.github.io/aima-exercises/robotics-exercises/#FigEx2). 1. Consider the robot configurations shown in Figure [FigEx2](https://aimacode.github.io/aima-exercises/robotics-exercises/#FigEx2)(a) through (c), ignoring the obstacle shown in each of the diagrams. Draw the corresponding arm configurations in configuration space. (*Hint:* Each arm configuration maps to a single point in configuration space, as illustrated in Figure [FigArm1](https://aimacode.github.io/aima-exercises/robotics-exercises/#FigEx2)(b).)  
2. Draw the configuration space for each of the workspace diagrams in Figure [FigEx2](https://aimacode.github.io/aima-exercises/robotics-exercises/#FigEx2)(a)–(c). (*Hint:* The configuration spaces share with the one shown in Figure [FigEx2](https://aimacode.github.io/aima-exercises/robotics-exercises/#FigEx2)(a) the region that corresponds to self-collision, but differences arise from the lack of enclosing obstacles and the different locations of the obstacles in these individual figures.)  
3. For each of the black dots in Figure [FigEx2](https://aimacode.github.io/aima-exercises/robotics-exercises/)(e)–(f), draw the corresponding configurations of the robot arm in workspace. Please ignore the shaded regions in this exercise.  
4. The configuration spaces shown in Figure [FigEx2](https://aimacode.github.io/aima-exercises/robotics-exercises/#FigEx2)(e)–(f) have all been generated by a single workspace obstacle (dark shading), plus the constraints arising from the self-collision constraint (light shading). Draw, for each diagram, the workspace obstacle that corresponds to the darkly shaded area.  
5. Figure [FigEx2](https://aimacode.github.io/aima-exercises/robotics-exercises/#FigEx2)(d) illustrates that a single planar obstacle can decompose the workspace into two disconnected regions. What is the maximum number of disconnected regions that can be created by inserting a planar obstacle into an obstacle-free, connected workspace, for a 2DOF robot? Give an example, and argue why no larger number of disconnected regions can be created. How about a non-planar obstacle?

**(a)**

**(b)**

**(c)**

**(d)**

**(e)**

**(f)**

[Exercise 9](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_9/)

Consider a mobile robot moving on a horizontal surface. Suppose that the robot can execute two kinds of motions:  
- Rolling forward a specified distance.  
- Rotating in place through a specified angle.  
The state of such a robot can be characterized in terms of three parameters ⟨x,y,ϕ⟨x,y,ϕ, the x-coordinate and y-coordinate of the robot (more precisely, of its center of rotation) and the robot’s orientation expressed as the angle from the positive x direction. The action “Roll(D)Roll(D)” has the effect of changing state ⟨x,y,ϕ⟨x,y,ϕ to ⟨x+Dcos(ϕ),y+Dsin(ϕ),ϕ⟩⟨x+Dcos⁡(ϕ),y+Dsin⁡(ϕ),ϕ⟩, and the action Rotate(θ)Rotate(θ) has the effect of changing state  
⟨x,y,ϕ⟩⟨x,y,ϕ⟩ to ⟨x,y,ϕ+θ⟩⟨x,y,ϕ+θ⟩. 1. Suppose that the robot is initially at ⟨0,0,0⟩⟨0,0,0⟩ and then executes the actions Rotate(60∘)Rotate(60∘), Roll(1)Roll(1), Rotate(25∘)Rotate(25∘), Roll(2)Roll(2). What is the final state of the robot?  
2. Now suppose that the robot has imperfect control of its own rotation, and that, if it attempts to rotate by θθ, it may actually rotate by any angle between θ−10∘θ−10∘ and θ+10∘θ+10∘. In that case, if the robot attempts to carry out the sequence of actions in (A), there is a range of possible ending states. What are the minimal and maximal values of the x-coordinate, the y-coordinate and the orientation in the final state?  
3. Let us modify the model in (B) to a probabilistic model in which, when the robot attempts to rotate by θθ, its actual angle of rotation follows a Gaussian distribution with mean θθ and standard deviation 10∘10∘. Suppose that the robot executes the actions Rotate(90∘)Rotate(90∘), Roll(1)Roll(1). Give a simple argument that (a) the expected value of the location at the end is not equal to the result of rotating exactly 90∘90∘ and then rolling forward 1 unit, and (b) that the distribution of locations at the end does not follow a Gaussian. (Do not attempt to calculate the true mean or the true distribution.)  
The point of this exercise is that rotational uncertainty quickly gives rise to a lot of positional uncertainty and that dealing with rotational uncertainty is painful, whether uncertainty is treated in terms of hard intervals or probabilistically, due to the fact that the relation between orientation and position is both non-linear and non-monotonic.

**Simplified robot in a maze. See Exercise**[**robot-exploration-exercise**](https://aimacode.github.io/aima-exercises/robotics-exercises/)

[Exercise 10 (robot-exploration-exercise)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_10/)

Consider the simplified robot shown in Figure [FigEx3](https://aimacode.github.io/aima-exercises/robotics-exercises/#FigEx3). Suppose the robot’s Cartesian coordinates are known at all times, as are those of its goal location. However, the locations of the obstacles are unknown. The robot can sense obstacles in its immediate proximity, as illustrated in this figure. For simplicity, let us assume the robot’s motion is noise-free, and the state space is discrete. Figure [FigEx3](https://aimacode.github.io/aima-exercises/robotics-exercises/#FigEx3) is only one example; in this exercise you are required to address all possible grid worlds with a valid path from the start to the goal location.  
1. Design a deliberate controller that guarantees that the robot always reaches its goal location if at all possible. The deliberate controller can memorize measurements in the form of a map that is being acquired as the robot moves. Between individual moves, it may spend arbitrary time deliberating.  
2. Now design a *reactive* controller for the same task. This controller may not memorize past sensor measurements. (It may not build a map!) Instead, it has to make all decisions based on the current measurement, which includes knowledge of its own location and that of the goal. The time to make a decision must be independent of the environment size or the number of past time steps. What is the maximum number of steps that it may take for your robot to arrive at the goal?  
3. How will your controllers from (a) and (b) perform if any of the following six conditions apply: continuous state space, noise in perception, noise in motion, noise in both perception and motion, unknown location of the goal (the goal can be detected only when within sensor range), or moving obstacles. For each condition and each controller, give an example of a situation where the robot fails (or explain why it cannot fail).

[Exercise 11 (subsumption-exercise)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_11/)

In Figure [Fig5](https://aimacode.github.io/aima-exercises/robotics-exercises/)(b) on page , we encountered an augmented finite state machine for the control of a single leg of a hexapod robot. In this exercise, the aim is to design an AFSM that, when combined with six copies of the individual leg controllers, results in efficient, stable locomotion. For this purpose, you have to augment the individual leg controller to pass messages to your new AFSM and to wait until other messages arrive. Argue why your controller is efficient, in that it does not unnecessarily waste energy (e.g., by sliding legs), and in that it propels the robot at reasonably high speeds. Prove that your controller satisfies the dynamic stability condition given on page [polygon-stability-condition-page](https://aimacode.github.io/aima-exercises/robotics-exercises/).

[Exercise 12 (human-robot-exercise)](https://aimacode.github.io/aima-exercises/robotics-exercises/ex_12/)

(This exercise was first devised by Michael Genesereth and Nils Nilsson. It works for first graders through graduate students.) Humans are so adept at basic household tasks that they often forget how complex these tasks are. In this exercise you will discover the complexity and recapitulate the last 30 years of developments in robotics. Consider the task of building an arch out of three blocks. Simulate a robot with four humans as follows:  
**Brain.** The Brain direct the hands in the execution of a plan to achieve the goal. The Brain receives input from the Eyes, but *cannot see the scene directly*. The brain is the only one who knows what the goal is.  
**Eyes.** The Eyes report a brief description of the scene to the Brain: “There is a red box standing on top of a green box, which is on its side” Eyes can also answer questions from the Brain such as, “Is there a gap between the Left Hand and the red box?” If you have a video camera, point it at the scene and allow the eyes to look at the viewfinder of the video camera, but not directly at the scene.  
**Left hand** and **right hand.** One person plays each Hand. The two Hands stand next to each other, each wearing an oven mitt on one hand, Hands execute only simple commands from the Brain—for example, “Left Hand, move two inches forward.” They cannot execute commands other than motions; for example, they cannot be commanded to “Pick up the box.” The Hands must be *blindfolded*. The only sensory capability they have is the ability to tell when their path is blocked by an immovable obstacle such as a table or the other Hand. In such cases, they can beep to inform the Brain of the difficulty.

26. Philosophical Foundations

[Exercise 1](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_1/)

Go through Turing’s list of alleged “disabilities” of machines, identifying which have been achieved, which are achievable in principle by a program, and which are still problematic because they require conscious mental states.

[Exercise 2](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_2/)

Find and analyze an account in the popular media of one or more of the arguments to the effect that AI is impossible.

[Exercise 3](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_3/)

Attempt to write definitions of the terms “intelligence,” “thinking,” and “consciousness.” Suggest some possible objections to your definitions.

[Exercise 4](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_4/)

Does a refutation of the Chinese room argument necessarily prove that appropriately programmed computers have mental states? Does an acceptance of the argument necessarily mean that computers cannot have mental states?

[Exercise 5 (brain-prosthesis-exercise)](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_5/)

In the brain replacement argument, it is important to be able to restore the subject’s brain to normal, such that its external behavior is as it would have been if the operation had not taken place. Can the skeptic reasonably object that this would require updating those neurophysiological properties of the neurons relating to conscious experience, as distinct from those involved in the functional behavior of the neurons?

[Exercise 6](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_6/)

Suppose that a Prolog program containing many clauses about the rules of British citizenship is compiled and run on an ordinary computer. Analyze the “brain states” of the computer under wide and narrow content.

[Exercise 7](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_7/)

Alan Perlis [[Perlis:1982](https://aimacode.github.io/aima-exercises/philosophy-exercises/)] wrote, “A year spent in artificial intelligence is enough to make one believe in God”. He also wrote, in a letter to Philip Davis, that one of the central dreams of computer science is that “through the performance of computers and their programs we will remove all doubt that there is only a chemical distinction between the living and nonliving world.” To what extent does the progress made so far in artificial intelligence shed light on these issues? Suppose that at some future date, the AI endeavor has been completely successful; that is, we have build intelligent agents capable of carrying out any human cognitive task at human levels of ability. To what extent would that shed light on these issues?

[Exercise 8](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_8/)

Compare the social impact of artificial intelligence in the last fifty years with the social impact of the introduction of electric appliances and the internal combustion engine in the fifty years between 1890 and 1940.

[Exercise 9](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_9/)

I. J. Good claims that intelligence is the most important quality, and that building ultraintelligent machines will change everything. A sentient cheetah counters that “Actually speed is more important; if we could build ultrafast machines, that would change everything,” and a sentient elephant claims “You’re both wrong; what we need is ultrastrong machines.” What do you think of these arguments?

[Exercise 10](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_7/)

Analyze the potential threats from AI technology to society. What threats are most serious, and how might they be combated? How do they compare to the potential benefits?

[Exercise 11](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_11/)

How do the potential threats from AI technology compare with those from other computer science technologies, and to bio-, nano-, and nuclear technologies?

[Exercise 12](https://aimacode.github.io/aima-exercises/philosophy-exercises/ex_12/)

Some critics object that AI is impossible, while others object that it is *too* possible and that ultraintelligent machines pose a threat. Which of these objections do you think is more likely? Would it be a contradiction for someone to hold both positions?