

CMPT 310, Spring 2019, Written Assignment 01

Due date: January 14, 2019

Problem 1. Suppose that the performance measure is concerned with just the first T 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 is has reached.

Solution . This question tests the students understanding of environments, rational actions, and performance measures. Any sequential environment in which rewards may take time to arrive will work, because then we can arrange for the reward to be over the horizon. Suppose that in any state there are two action choices, a and b , and consider two cases: the agent is in state s at time T or at time $T - 1$. In state s , action a reaches state s with reward 0, while action b reaches state s again with reward 1; in s either action gains reward 10. At time $T - 1$, its rational to do a in s , with expected total reward 10 before time is up; but at time T , its rational to do b with total expected reward 1 because the reward of 10 cannot be obtained before time is up. Students may also provide common-sense examples from real life: investments whose payoff occurs after the end of life, exams where it doesnt make sense to start the high-value question with too little time left to get the answer, and so on.

The environment state can include a clock, of course; this doesnt change the gist of the answer now the action will depend on the clock as well as on the non-clock part of the state but it does mean that the agent can never be in the same state twice

Problem 2. For each of the following activities, give a PEAS description of the task environments and characterize it in terms of properties.

(a) Exploring the subsurface oceans of Titan.

Solution . Partially observable, stochastic, sequential, dynamic, continuous, single agent (unless there are alien life forms that are usefully modeled as agents).

(b) Playing a tennis match.

Solution . Fully observable, stochastic, episodic (every point is separate), dynamic, continuous, multi-agent.

(c) Practicing tennis against a wall.

Solution . Fully observable, stochastic, episodic, dynamic, continuous, single agent.

- (d) Knitting a sweater.

Solution . Fully observable, deterministic, sequential, static, continuous, single agent.

Problem 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.

- (a) Formulate this problem. How large is the state space?

Solution . Well define the coordinate system so that the center of the maze is at $(0, 0)$, and the maze itself is a square from $(-1, -1)$ to $(1, 1)$.

Initial state: robot at coordinate $(0, 0)$, facing North.

Goal test: either $|x| > 1$ or $|y| > 1$ where (x, y) is the current location.

Successor function: move forwards any distance d ; change direction robot it facing.

Cost function: total distance moved.

- (b) In navigation 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?

Solution . The state will record the intersection the robot is currently at, along with the direction its facing. At the end of each corridor leaving the maze we will have an exit node. Well assume some node corresponds to the center of the maze.

Initial state: at the center of the maze facing North.

Goal test: at an exit node.

Successor function: move to the next intersection in front of us, if there is one; turn to face a new direction.

Cost function: total distance moved.

There are $4n$ states, where n is the number of intersections.

- (c) 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?

Solution . Initial state: at the center of the maze.

Goal test: at an exit node.

Successor function: move to next intersection to the North, South, East, or West.

Cost function: total distance moved.

We no longer need to keep track of the robots orientation since it is irrelevant to predicting the outcome of our actions, and not part of the goal test. The motor system that executes this plan will need to keep track of the robots current orientation, to know when to rotate the robot.

- (d) 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.

Solution . State abstractions:

- (i) Ignoring the height of the robot off the ground, whether it is tilted off the vertical.
- (ii) The robot can face in only four directions.
- (iii) Other parts of the world ignored: possibility of other robots in the maze, the weather in the Caribbean.

Action abstractions:

- (i) We assumed all positions we safely accessible: the robot couldnt get stuck or damaged.
- (ii) The robot can move as far as it wants, without having to recharge its batteries.
- (iii) Simplified movement system: moving forwards a certain distance, rather than controlled each individual motor and watching the sensors to detect collisions.

Problem 4. Give a complete problem formulation for each of the following. Choose a formulation that is precise enough to be implemented.

- (a) Using only four colors, you have to color a planar map in such a way that no two adjacent regions have the same color.

Solution . Initial state: No regions colored.

Goal test: All regions colored, and no two adjacent regions have the same color.

Successor function: Assign a color to a region.

Cost function: Number of assignments.

- (b) 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.

Solution . Initial state: As described in the text.

Goal test: Monkey has bananas.

Successor function: Hop on crate; Hop off crate; Push crate from one spot to another; Walk from one spot to another; grab bananas (if standing on crate).

Cost function: Number of actions.

- (c) 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.

Solution . Initial state: considering all input records.

Goal test: considering a single record, and it gives illegal input message.

Successor function: run again on the first half of the records; run again on the second half of the records.

Cost function: Number of runs.

Note: This is a contingency problem; you need to see whether a run gives an error message or not to decide what to do next.

- (d) 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.

Solution . Initial state: jugs have values $[0, 0, 0]$.

Successor function: given values $[x, y, z]$, generate $[12, y, z]$, $[x, 8, z]$, $[x, y, 3]$ (by filling); $[0, y, z]$, $[x, 0, z]$, $[x, y, 0]$ (by emptying); or for any two jugs with current values x and y , pour y into x ; this changes the jug with x to the minimum of $x + y$ and the capacity of the jug, and decrements the jug with y by the amount gained by the first jug.

Cost function: Number of actions.

Problem 5. Considering the decision of not exploring negative path costs, answer the following items.

- (a) Suppose that actions can have arbitrarily large negative costs: explain why this possibility would force any optimal algorithm to explore the entire state space.

Solution . Any path, no matter how bad it appears, might lead to an arbitrarily large reward (negative cost). Therefore, one would need to exhaust all possible paths to be sure of finding the best one.

- (b) Does it help if we insist that step costs must be greater than or equal to some negative constant c ? Consider both trees and graphs.

Solution . Suppose the greatest possible reward is c . Then if we also know the maximum depth of the state space (e.g. when the state space is a tree), then any path with d levels remaining can be improved by at most cd , so any paths worse than cd less than the best path can be pruned. For state spaces with loops, this guarantee doesn't help, because it is possible to go around a loop any number of times, picking up c reward each time.

- (c) 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?

Solution . The agent should plan to go around this loop forever (unless it can find another loop with even better reward).

- (d) 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 sections for route finding so the artificial agents can also avoid looping.

Solution . The value of a scenic loop is lessened each time one revisits it; a novel scenic sight is a great reward, but seeing the same one for the tenth time in an hour is tedious, not rewarding. To accommodate this, we would have to expand the state space to include a memory a state is now represented not just by the current location, but by a current location and a bag of already-visited locations. The reward for visiting a new location is now a (diminishing) function of the number of times it has been seen before.

- (e) Can you think of a real domain in which step costs are such as to cause looping?

Solution . Real domains with looping behavior include eating junk food and going to class.