Eduardo Gutarra Velez
Foundations of Artificial Intelligence
Feb 8, 2010

Assignment 4

1. a. Considering the facts that the agent knows its geography and doesn't know either its starting location or whether there is dirt there. The simple vacuum-cleaner agent will begin a sequence of actions that could be:

Move to the next possible square if the current one is clean by using its two possible actions of moving left or right. This depends on its starting location that is not known, but it knows where it is located once it is put to perform.
If it is dirty suck the dirt from the square.

The performance measure as it is described for figure 2.3 awards a point for each clean square at each time step for a period of 1000 time steps.

Given its percept sequence it has performed the correct actions, the actions that achieve it more points. Also it must be taken into account that the agent can only clean a square if it stands on it, so even though cleaning a dirty square without visiting it as soon as it becomes dirty would achieve a higher score. It is not a possible action that the agent can perform.

So given the agent's prior knowledge, actions it can perform and its percept sequence, it has effectively behaved such that it is keeping the squares clean as best it can maximizing the performance measure, and therefore being rational.

1.b. Given the fact that clean squares stay clean and sucking cleans the current square the rational function for the described case would be for the agent to:
Move to the right or left if the current square is clean and the next square (right or left) has not been visited yet.
Clean the dirt if the current square is dirty.
Do nothing if the current square is clean and the next square (right or left) has already been visited.
For this function you would require your agent program to keep an internal state that recalls whether it has visited the A or B square before, depending where it starts.

1.c. Because the agent starts without knowledge of the geography of its environment. It will need explore it and determine where the bounds are. In order to do this, it needs to keep an internal state so that it can keep track of the squares it has visited. If it lacks this internal state, the agent might be visiting the same squares over and over again leaving unexplored squares in geographies where there are more than 2 squares. If the squares get dirty following a pattern, then it is possible to have an agent with planning capabilities track it. By doing so, it can find the most efficient way to clean the squares as they get dirty. However, if the squares get dirty stochastically the agent just needs to ensure it explores every square methodically, so that it doesn't leave squares unexplored.

2. The performance measure is incorporated by the designer of the Agent. It is basically an objective criterion for evaluating an agent's degree of success after performing. This measure may also be used by the agent itself as feedback in order to learn or improve its actions, so that it may increase its degree of success later on.
The utility is instead, a function that maps each possible outcome state to a value (utility) that describes a degree of happiness (or closeness to a goal). The utility function may also handle the tradeoff for conflicting goals that may be encountered in the outcome states. A utility-based agent uses this function in order to choose the action with the highest expected utility , which hopefully takes it to closer to its goal.

3.a. False. If an agent knew everything about the state of its environment it would be omniscient. It is possible for an agent to be perfectly rational without knowing everything. For example, the taxi driver agent can be perfectly rational without knowing what the other drivers are thinking or what their next move is going to be.

3.b. True. An example of how a simple pure reflex agent cannot behave rationally in a given task environment is in question 1.b, where in order for the agent to maximize its performance measure, it must remember what squares it has visited. Because the pure reflex agent cannot remember what squares it has visited, there is no way to design it to maximize its performance measure, and therefore it cannot behave rationally.

3.c. True. If you have a task environment with a performance measure that awards nothing for any percept history, you can have any agent perform any behavior, and it will be rational, because it is always at the maximum performance measure.

3.d. True. You can have an agent perform essentially the same optimal actions in environments where their differences do not change what the right action would be. A simple example could be an agent that explores the surface of solid planets (i.e. mars, earth).

3.e. False. You can have a performance measure in which your agent requires information from the environment in order to maximize it. The agent cannot obtain this information because the environment is unobservable, and therefore it cannot be rational.

4.

Task Environment	Performance Measure	Environment	Actuators	Sensors
Playing soccer	Maximize net game score = team score – opponent score	Soccer field, opponent players, allies, referee, soccer ball	A form of locomotion and a mechanism for kicking the ball. Transmitter to communicate	Visual sensor, Touch sensors, and Signal receivers.

Task Environment	Performance Measure	Environment	Actuators	Sensors
Exploring the subsurface oceans of Titan	Coverage area of exploration and number of different things found.	Ocean surface and subsurface & obstacles.	Propeller to move forward, engine, steering to turn where necessary Sonar transmitter	Sonar Receiver, cameras, chemical sensors, distance sensors.
Shopping for used AI Books on the Internet	Finding lowest price for a given AI book.	Shopping search engines, User.	Protocol to communicate with search engines	Protocol to receive information from search engines
Playing a tennis match	Maximize net game score = agent's score – opponent score	Tennis court with referee, tennis ball, opponent and racket.	A form of locomotion and a mechanism for hitting the ball with the racket	Visual sensor, Touch sensors.
Practicing tennis against a wall	Maximize the number of hits against a wall without Dropping the ball.	Wall, tennis ball and racket.	A form of locomotion and a mechanism for hitting the ball with the racket	Visual sensor, Touch sensors
Performing a High Jump	Maximize jump height	Bar that sets the height, space to run before performing the jump.	A form of locomotion and a mechanism to performing the jump.	Visual sensors, Touch sensors, angle joint sensors.
Knitting a sweater	Consistency of knitting and correct measurements.	Yarn, and knitting needles	Robotic joints to perform the knitting.	Visual sensors, Touch sensors.
Bidding on an item at an auction	Getting the lowest price for an item without exceeding a user defined price.	Different auctioneers for the same item, other bidders, user defined price, and time limitation for auction.	Form of communicating a bid to the auctioneers.	Sensors to collect the current bids on an item with different auctioneers

Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Playing soccer	Partially	Stochastic	Sequential	Dynamic	Continuous	Multiple
Exploring the subsurface oceans of Titan	Partially Observable	Stochastic	Sequential	Dynamic	Continuous	Single or Multiple
Shopping for used AI Books on the Internet	Fully Observable	Deterministic	Sequential	Dynamic	Discrete	Single
Playing a tennis match	Partially Observable	Stochastic	Sequential	Dynamic	Continuous	Single
Practicing tennis against a Wall	Fully Observable	Deterministic	Sequential	Static	Continuous	Single
Performing a High Jump	Fully Observable	Deterministic	Episodic	Static	Continuous	Single
Knitting a sweater	Fully Observable	Deterministic	Sequential	Static	Discrete	Single or Multiple
Bidding on an item at an auction	Partially Observable	Stochastic	Sequential	Dynamic	Continuous	Single