Artificial Intelligence (CS 534)
Assignment 1 submission

1.

a) **False**. Even though the Vacuum cleaning agent from is unaware of the status of the adjacent square, it is still acting rationally. Perfect rationality refers to the ability to make good decisions given the sensor information received. An agent that only senses partial information could be rational: consider an agent whose performance measure is based on its ability to stay completely static in an environment where there are no other agents. Such an agent could be perfectly rational even if it did not have any sensors at all. Granted, such an agent would not be very interesting, but it would indeed be rational.

b) **True**. Consider a partially unobservable task environment where the performance measure is based on the agent's extent of exploration. Now assume that this agent is purely reflex, only responding when it's world state changes. Since its performance metric is to explore but its actions are conditional upon change around it. The agent waits for its state to change but it never will because it is the agent's entire job is to change its state.

c) **True**. Consider an environment that disables all agent's actuators, rendering them immobile. Regardless of what performance measures these agents possessed, they are still judged as rational because they are physically incapable of performing anything.

d) **False**. The input to the agent function is the entire percept sequence up to a that point whereas the input to the agent program is the input sensed by the sensors.

e) **False**. An agent function is defined as a map of any given percept sequence to an action. Consider an agent who has no sensors and thereby has no percept sequence. This would imply that the domain of the agent function does not exist which contradicts the definition of the agent function. Therefore, there are program/machine combinations that cannot implement agent functions.

f) **True**. Consider a task environment where it is a multi-agent, competitive and, thereby, deterministic game environment. Now consider an agent whose performance measure is based on its ability to be unpredictable to its enemies. An agent selecting an action uniformly from the set of possible actions would, therefore, be very rational as it would be difficult for another agent program to predict what the agent will do.

g) **True**. Aside from the unattainability of perfect rationality, consider an agent whose performance metric is based on two things: the ability to escape from enemies chasing it and the ability to stay underwater for long periods of time. Further assume that this agent has sensors and actuators that give it the ability to excel at these two things and they do not affect each other or the agent adversely. Now consider two distinct task environments: one is a multi-agent, competitive environment in which only physical means are employed to defeat another agent and the other is a competition to stay underwater the longest. The agent described above would be perfectly rational in both environments.

h) **False**.

i) **False**. Due to the partially observable, multi-agent, stochastic, sequential, static and discrete nature of poker, even a perfectly rational agent could not possibly win every game unless the agent is completely omniscient, there are too many factors contributing to the complexity of the task environment that are beyond the agent's scope of influence

3.

3a.

The initial state of this robot is the centre of the maze and it is facing north. The robot can face north, south, east or west and can move a certain distance. It will stop if it comes too close to a wall. Since the robot can travel a certain distance and can face 4 directions, it can move in those 4 directions as well. Given this, with any direction the robot moves in, it will either stop upon reaching a wall to avoid collision, or progress further through the maze.

The robot has a total of five possible actions: face north, face south, face east, face west, and move. It can move in the direction it is facing unless it reaches a wall.

To formulate the problem, we should start by setting up xy coordinate frame. Let's define the coordinate frame so that the centre of the maze is at (0,0), and the maze can be a square from (-1, -1) to (1, 1).

Initial state: Robot at middle of the maze, at coordinates (0, 0), facing North

Actions: Face north, face south, face east, face west, move

Successor function: move forward any distance d; change direction robot is facing to one of the other directions

Goal test: either $|x| > 1$ or $|y| > 1$ where (x, y) is the current location of the robot. Since the maze is of unit length, if modulus of either of coordinate x or y goes beyond 1, that implies the robot is out of the maze.

Cost function: Total cost moved

The state space is infinitely large, since the robot's position is continuous.

3b.

Now that we can only turn at the intersection of two or more corridors, we need to have an exit node at the end of each corridor. Otherwise, we won't be able to reach the target. The initial state will be in the centre of the maze facing north. The test will be to get to an exit node. The successor function is to move past the intersection if there is one in front of us, and the cost function is just the total distance moved as before.

Initial state: Robot at the centre of the maze facing North.

Goal test: Robot at an exit node.

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

Cost function: Total distance moved by robot.

The state space is no longer infinite as before and will have changed due to the number of intersections. Therefore, the state space is now 4N with N being the number of intersections.

3c.

The direction won't matter since we're just looking to reach a wall. We don't need to keep track of the robot's orientation since it has no relevance in predicting the outcome of our actions. Hence, it is not a part of the goal test.

Initial state: Robot at the centre of the maze.

Goal test: Robot at an exit node.

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

Cost function: Total distance covered.

3d.

Three of the many simplifications that we made are as follows:

1) We assumed that the robot can only face 4 directions.

2) We assumed that all positions are safely accessible i.e., the robot could not get stuck or damaged.

3) We also ignored possibility of other robots in the same area or other items in the way of the robot moving in its "space".

4.

4a.

This search problem consists of the following components:

Goal: Place both people, i.e. p1 and p2 in the same city i, in the lowest possible time T.

D(i, j) as the straight-line distance between two cities i and j.

State space: All possible city pairs (i, j) where both people p1 and p2 can be at. i=j if both are in the same city.

Successor function: The successors of (i, j) are all pairs (x, y) such that Adjacent(x, i) and Adjacent(y, j).

Goal: Be at (i, i) for some i.

Step cost function: Moving p1 to city i' and p2 to city j', incurring a cost of Max(D(i, i'), D(j, j'))

4b.

In the ideal case, both people proceed toward each other in steps of equal size, resulting in twice the distance reduction compared to if one of them had simply walked to the other. Therefore, the solution becomes more optimal as it approaches the heuristic. Therefore, of the heuristic functions mentioned, heuristic function D(i, j)/2 is admissible.

4c.

Yes. There are completely connected maps for which no solution exists. Consider a map with just two cities, with the friends starting in different cities. They are bound to move back and forth, never meeting.

4d.

Yes. If the friends start at locations A and B, they can meet at another location K, and that requires B to go once around the loop and come back to B.