Name: Abhishek Basu
UIN: 658369163

CS 411 - Artificial Intelligence I
Fall 2021
Assignment 1
Department of Computer Science, University of Illinois at Chicago

Total Points: 25

1. Explain why the "Part-Picking robot" environment (Fig 2.6, page 45, AIMA) is characterized with following properties - partially observable, single-agent, stochastic, episodic, dynamic and continuous. [6*1]
   Ans:
   - Partially Observable: According to the part-picking robot environment, the robot does not have full visibility of the entire area. It might provide partial information about the location and the condition because of limited sensors or cameras. For example, if the robot depends on a camera that can only see a part of the area at a time, it won't have a full image of the environment at all the time.
   - Single-Agent: This environment is designed for a single robot, which operates single-handedly to achieve its goals, like picking up parts and placing them in accurate place. There are no other agents working within this environment except this single robot.
   - Stochastic: It refers to the uncertainties involved in the actions performed by robot and environment's response. For example, the robot may not always pick up the correct one due to inaccuracies in its sensors or their moves are also unpredictable due to vibration and other obstacles or disturbances etc.
   - Episodic: The environment is episodic if each picking task or episode is not depended on the previous one. Each action of part picking and placing it in a new place can be considered as a different episode, where the output does not directly affect the next step in a meaningful way, except the state of the environment which may change.
   - Dynamic: When the state of the environment can change over the period, even if the robot is not actively engaging with it. For example, parts may shift or move slightly, and robot must adapt to these changes as it performs its tasks.
   - Continuous: The environment is continuous in terms of space and possible time. The robot can move to different range of positions within the workplace and make relatable adjustments in its actions.
2. Give PEAS description for the robotic-soccer environment. [2*4]
   Ans:
   - Performance Measure: score, accuracy, efficiency, defensive and offensive success.
   - Environment: Soccer field, Soccer ball, opponent team, team members, rules, obstacles, audience.
   - Actuators: legs of robot, motors, kick mechanism, grippers or manipulators, navigator.
   - Sensors: cameras, proximity sensors, gyroscopes and accelerometers, ball detection sensors

3. Define rational agent and autonomous agent in your own words [4]
   Ans:
   - Rational Agent
     A rational agent is an entity that makes decisions based on the goal of maximizing its performance in terms of success. In other words, their decision depends on predetermined set of rules and current state of environment.
     Key aspects: goal oriented- it acts in a particular way so that it aligns with its goals and objectives

Informed decision making- it uses the information is available to make the best possible decisions given its current knowledge
Optimality- it makes great efforts to achieve the best possible output from its actions, under these conditions.

For example: a rational agent in a chess game knows the best moves to maximize its chances for winning the game, with all possible moves and strategies.

- Autonomous Agent
  It's a type of agent that operates independently and make decision without any constant human intervention. It is designed in a way that it can perceive and sense its environment s own judgement and take actions to complete the tasks.
  Key aspects: Independence- it can perform tasks and make decision on its own without any external control.
  Self-Sufficiency- It is capable to handle tasks and respond to its environment independently.
  Adaptability- It can adjust its behavior based on changes in its environment or internal state

For example: a self-driving car is an autonomous agent because it can navigate, make driving decisions and adjust its behavior based on the obstacles (road conditions) and traffic without needing external control (human driver).

4. You have the agent program for simple reflex agent as shown below

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action
  if *status* = *Dirty* **then return** *Suck*
  else if *location* = *A* **then return** *Right*
  else if *location* = *B* **then return** *Left*

Figure 2.8 The agent program for a simple reflex agent in the two-state vacuum environment. This program implements the agent function tabulated in Figure 2.3.

Now suppose you can use one more action "NoOp" which does nothing.

a. Would it be desirable to use this action in this agent program to prevent vacuum cleaner from moving after all squares are cleaned? If yes, modify the simple reflex agent program adding "NoOp" action and if not give an argument that despite the option of "NoOp" the agent is bound to keep moving. [4]
Ans:
Using "NoOp" in the simple reflex agent
Current behavior: the agent moves left or right depending on the location A and B but does not account for whether all squares have been cleaned.
Even if both squares are clean the agent will keep moving based on its location rules ('right' if location A, 'left' if at location B).
Adding "NoOp"
Purpose: allowing the agent to stop moving when both squares are clean.
Modified program:
Function REFLEX-VACUUM-AGENT ([location, status]) returns an action
  if status = Dirty then return Suck
  else if location = A then return Right
  else if location = B then return Left

else return NoOp

Argument:
regardless of adding “NoOp” the agent will only goint to return ‘NoOp’if it perceives no dirt at the current position and has no applicable movement rules. However, since the agent’s actions are depicted by current percepts, and it requires an internal state or memory of cleaning all the squares it may not stop moving efficiently
Example: if the agent is at A and square is clean it will move right. If it, then ends up in B with a clean square too it will move left. The agent can’t recognize when it has completed the task of cleaning the particular area without internal state.

b. If you have a choice of using other type of agents, which one would you choose and how would it prevent vacuum from moving after all squares are cleaned? [3]
Ans:
I would choose “Model – based Reflex Agent” because its internal state maintains a record of cleaned squares and the agent’s location and its goal is to stop moving once all squares are confirmed clean.
Behavior: If the current square is dirty, it will ‘suck’ and update the internal state to mark the area as clean.
If the current square is clean: checking the memory to determine if there are any unclean squares are there to clean.
If there are still unclean squares just move (‘Left’ or ‘right’) following the internal map.
If all squares are clean: Execute ‘NoOp’ (to stop moving).
Advantages
Memory: it tracks which areas have been cleaned so it can make accurate decision about when to stop.
Efficiency: this approach ensures that robot does not continue moving once it has done cleaning all the squares. this improves performance as it won’t waste time moving here and there unnecessarily

Use the assumptions listed below

- The performance measure awards one point for each clean square at each time step, over a “lifetime” of 1000-time steps.
- The “geography” of the environment is known a priori (Figure 2.2) but the dirt distribution and the initial location of the agent are not. Clean squares stay clean, and sucking cleans the current

square. The Left and Right actions move the agent left and right except when this would take the agent outside the environment, in which case the agent remains where it is.

- The only available actions are Left, Right, Suck and NoOp
- The agent correctly perceives its location and whether that location contains dirt