CSE 571 Fall 2022 HW1

Submitted By: Aman Peshin (1225476655)

Exercise 1.1

a. False. We know that the agent in this case does not know whether square B is dirty or not. Even without this information the simple reflex agent still moves from one location to another and gets the squares clean based on its condition-action rules. For this two-state environment, the agent is rational.
   Performance: clean all locations that are in the environment.
   Environment: dirt, location, vacuum
   Actuators: Left, Right, Suck, No-Op
   Sensors location sensor and dirt sensor

b. True. There exists a task environment that is defined by PEAS where no simple reflex agent can be rational. If we modify the sensors of the task environment, the simple reflex agent cannot sense whether there is dirt or not present in that particular location. It does not matter what agent we choose in this case, the performance measure cannot be satisfied.
   Performance: clean all locations that are in the environment.
   Environment: location, vacuum
   Actuators: Left, Right, Suck, No-Op
   Sensors: location sensor.

c. True. If we consider only one square as the environment for the agent. Then the agent will always satisfy its goal. It will suck up dirt if present, and even if it leaves traces behind, it will reach its goal of a clean square in the subsequent actions. Here we can create a task environment that is devoid of the location sensor.
   Performance: clean all locations that are in the environment.
   Environment: dirt, vacuum
   Actuators: Suck, No-Op
   Sensors: Dirt sensor

d. True. Let us take an example of 4 squares that are in a row. If there are two actions which the agent can select from i.e. suck and go right or suck and go left. Since the environment is deterministic the agent will take only its current percept. It will not know how many locations there are that need to be cleaned. It can still achieve its goal because the probability of going on either side is 50% and at random it can still clean all squares.
   Performance: clean all locations that are in the environment
   Environment: dirt, location, vacuum
   Actuators: Left, Right, Suck, No-Op
   Sensors: location sensor and dirt sensor

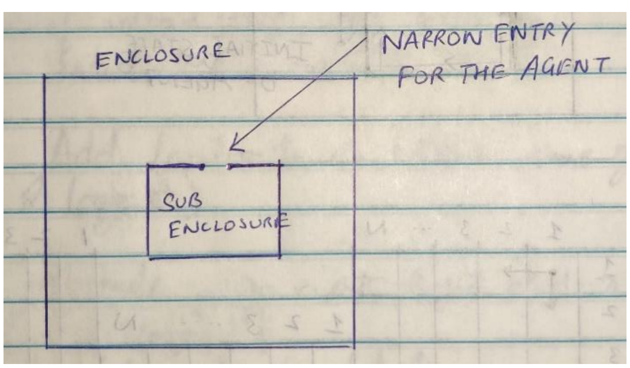
e. False, a vacuum agent without any built in knowledge of the environment cannot determine the topology of the space it needs to clean. Hence it can never achieve its goal of all clean locations. If the agent had built-in knowledge of the dirt and its corresponding location along with the total map of all the locations then it could be rational.
Performance: clean all locations that are in the environment
Environment: location, vacuum
Actuators: Left, Right, Suck, No-Op
Sensors location sensor, dirt sensor

Exercise 1.2

a. No, a simple reflex agent will not store any prior environment state internally and hence won't know if it has visited the same state or obstacle before. This agent is rational if the environment is simple and without obstacles. But since the environment is unobservable to the agent. There can be obstacles where the agent may get stuck in an infinite loop.

b. Yes, based on the previous answer if the contours of the environment are irregularly shaped, then a randomized action simple reflex agent is much better if it gets stuck in infinite loops. The advantage that this has over the simple agent is that if it visits the same state again in the infinite loop, due to its randomized behavior it may take a different action from the one it took before to escape this loop.

c. Let's take a penalty for every movement, and assume the environment to be a small enclosure inside another one, here we need to make sure that the entry to the sub enclosure is narrow enough only for the agent to move through. Here we see that if there is a narrow opening in the enclosure then the randomized action may take a lot of time to get through that passage and keep increasing its penalty while visiting already clean states or encountering the edges of the environment. Moreover, even if it enters the sub enclosure it may then turn back again from the state it came from, increasing the penalty even more(It turns back from the mouth of the entry point)

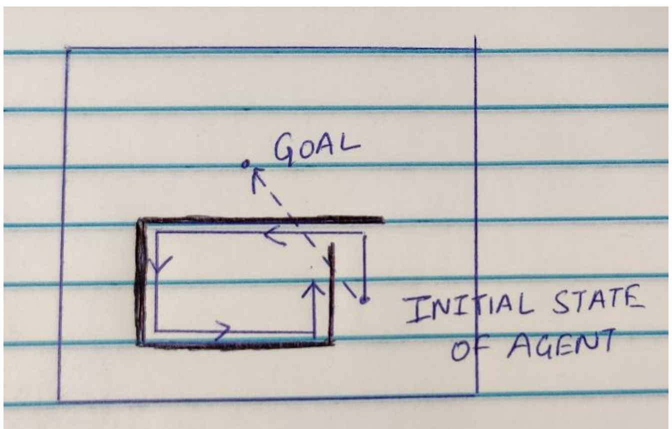


Exercise 1.3

a. The type of agent that best describes the above behavior is a simple reflex agent. Since it has only two condition actions we describe the pseudo-code as follows:

   function REFLEX-ACTION-AGENT([obstacle, relative_direction]) returns an action
   if obstacle detected then return (move left along contour)
   else return (move along relative_direction)

b. No, if the configuration of obstacles in the environment can change, this agent may not reach the goal in the following environment contour.



c. Yes we can use a model-based reflex agent to improve the agent. We would need a GPS sensor in order to help the agent come out of infinite loops it may get stuck in. Since a model-based reflex agent stores two parameters "model" and "internal state", it can based on previous percepts and GPS know what path it has taken. We would also need to amend the condition-action rule to change the direction if moving left along the contour. I have taken moving left along a contour/obstacle to be a preset and hard action of the agent.

Exercise 1.4

a. Complex metrics have various advantages and limitations and they depend on the type of environment the intelligent agent is deployed in, first let's look at the limitations that may arise by making the metric much more complex than it needs to be.
   1. If we assign multiple goals to an agent which don't actually simplify the task of the agent to reach its goal state, then we induce the problem that the agent will compromise its primary goal to satisfy the other non-important metrics to maximize them as well. This may cause the agent to be inefficient.
   2. Goals may be misinterpreted and could contribute to an inefficient agent. If we model streaming time as a metric for user satisfaction then we may have a recommendation agent that is bad at its job. These are consequences of setting big and complex metrics.
   3. Agents that have complex goals may also be computationally heavy, for simple applications use of these agents are inefficient as it may increase the cost of designing agents.

Some advantages of complex goals are as follows:

1. Having complex metrics allows the agent to satisfy its primary goal whilst maximizing other sub-goals as well. In the case of a self-driving car, the main goal is to reach the destination, but sub-goals may include cost, safety and distance traveled.
2. Complex metrics give a better and more complete picture of goal satisfaction than a simple yes/no output.
3. Complex goals can have some negative points/ penalize wrong actions of the agent, this way we can ensure that an agent does now maximize its performance metric like in the case where a vacuum agent sucks at clean squares to satisfy its goal. There can be a mix of penalty and bonus points for the different goals that are set.

b. No, this does not make a goal-based agent better than a simple reflex agent, if we set complex metrics to it. As referred to by the book we see that for the simple task of segregating items on a production line. A simple reflex agent will have finite segregation buckets into which it will pick and drop the items. If we set a goal-based agent in this episodic environment we will see that the agent will try to make sense of the distribution of different items it encounters or on the number of times a defective piece has come to it. This will in turn will update its knowledge on how it should classify the items and will alter its goals. In short, we do not want to introduce an advanced agent like a goal-based agent in a simple episodic environment.