

Aasish Tammana  
CSE 571 – Artificial Intelligence  
Homework 1

Exercise 1.1

a) False.

Performance Metric: Clean the room/(here tiles A and B)

Environment: Dirt, Location and Vacuum

Actuators: Left, Right, Suck, no operation.

Sensors: Location and current status of tiles

A partially observable agent does not track the state or care about the steps. It will just clean the current tile and move to the next tile. As per the performance metric both tiles must be clean. Hence it can be considered to be rational.

b) True.

Performance Metric: Clean the room/(here tiles A and B) at some cost

Environment: Dirt, Location and Vacuum

Actuators: Left, Right, Suck, no operation.

Sensors: Location and current status of tiles

A simple reflex agent just sucks the tile if it is dirty else moves to the other tile as per the agent function. It just takes decisions based on current input. So if performance involves a random action, since the agent cannot sense it, it will not perform well and cannot be called rational.

c) True.

Performance Metric: Clean the room/(assume single tile)

Environment: Dirt, Location and Vacuum

Actuators: Left, Right, Suck, no operation.

Sensors: Location and current status of tiles

Assume a scenario where every action of the agent is given a point irrespective of the action it takes. In this case cleaning the tile. The agent gets point if its fully cleaned or not. In this scenario it is rational.

d) True.

Performance Metric: Clean the room/(here tiles A and B)

Environment: Dirt, Location and Vacuum

Actuators: Left, Right, Suck, no operation.

Sensors: current status of tiles

As the agent knows the environment, it will follow a sequence of events based on the inputs. So if tile A is dirty it will suck and move on to tile B. Therefore, it is said to be rational in this environment.

e) False.

Performance Metric: Clean the room/(here tiles A and B)

Environment: Dirt, Location and Vacuum

Actuators: Left, Right, Suck, no operation.

Sensors: current status of tiles

Say the agent is present in tile A and it is dirty. It begins to suck and moves on to the next tile. But if it had only partially cleaned the tile it still doesn't know because the environment is unobservable. So it is unsuccessful in cleaning tile A and cannot be called rational.

## Exercise 1.2

a) No

Performance Metric: Clean the space

Environment: Dirt, Location, Obstacles and Vacuum

Actuators: Left, Right, Suck, no operation.

Sensors: current status of tiles

The agent cannot perceive its surroundings. So, if it goes to a corner and gets struck it will not be able to navigate to other tiles. Since the performance metric is to clean the space, the agent cannot be called rational in this case.

b) Yes

Performance Metric: Clean the space

Environment: Dirt, Location, Obstacles and Vacuum

Actuators: Left, Right, Suck, no operation.

Sensors: current status of tiles

In this case, the agent uses a randomised function. So even if it gets struck like in case “A” it would probably take an appropriate action which helps it overcome the situation. This agent would do well in case of minute spaces as probability is higher to cover the entire area through the randomness. Although, this could take a lot of time for bigger spaces, the chances of completing the task are higher than simple reflex agent.

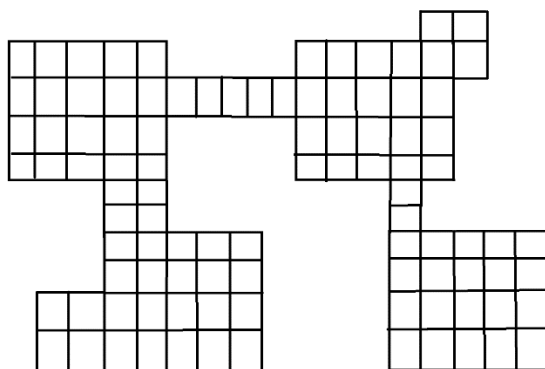
c)

Performance Metric: Clean the space, total movements

Environment: Dirt, Location, Obstacles and Vacuum

Actuators: Left, Right, Suck, no operation.

Sensors: current status of tiles



As per a randomized agent, since the environment is uniform, probability of the agent picking narrow paths is very less, almost inconsiderate. In this scenario the agent will not do well as every movement is valuable and the agent keeps revolving in the same locations.

### Exercise 1.3

Performance Metric: Reach destination (goal) from the start

Environment: Dirt, Location, Obstacles and Vacuum

Actuators: Left, Right, up, down

Sensors: Tactile

- a) This is the behaviour of a simple reflex agent which just follows a predefined set of actions.

This function can be written as:

Function Move\_To\_Goal([Obstacle,Location]) returns an action

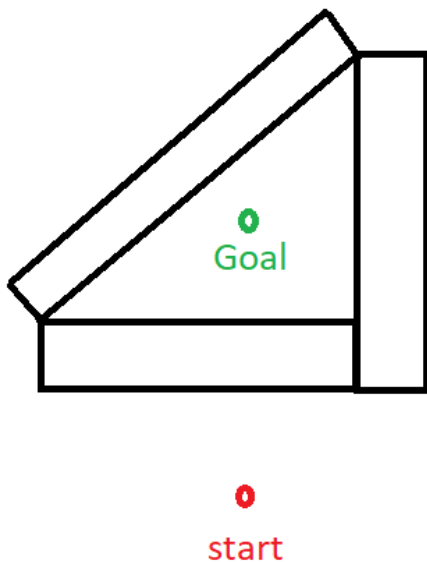
    If obstacle

        Move left

    Else if

        Move to goal

- b) No, if the obstacles form a closed loop, the agent will never be able to reach the goal. It will be stuck in an infinite loop around the obstacles as shown below.



However, if there exists a path, the agent follows the obstacles in counter clockwise direction with all possible movements up/down/left/right then it will be able to reach the goal as long as a path exists to the goal position.

- c) Yes, we can use a model based reflex agent. Let us assume that there is a shorter path to the right instead of left. If the agent can perceive the surroundings and remember its location, it could map the environment predict which path to take. In order to achieve this, the agent will need some location sensors.

#### Exercise 1.4

- a) Let us consider the two examples of a simple vacuum robot and the automated taxi. We know the PEAS for the Vacuum robot. In case of an automated taxi, PEAS can be defined as follows:

Performance Metric: Safety, Reaching Destination, Comfort

Environment: Traffic, Obstacles, Pedestrians, Other vehicles, Weather

Actuators: Brake, Accelerator, Lights, Display

Sensors: Accelerometers, GPS, Cameras, keyboard

If we compare the environment of an Automated Taxi, the characteristics are as follows. It is a partially observable, multi-agent, Stochastic environment, Sequential, Dynamic and Continuous.

In case of vacuum bot with simple metric of cleaning the tiles. The agent just has a predefined set of actions (table lookup) based on the current status of tiles i.e. suck or move. However, the automate taxi environment is only partially observable which means all relevant info is not available. How the other agents influence the agent's decision is unpredictable.

Hence a table lookup like a simple reflex agent is just not sufficient and it involves a high amount of predictive analysis. This is done by keeping track of previous actions/sequences. Basically, remember its past actions so that it can act accordingly. In order to check this, the agent needs to analyse what happens to the environment based on the actions it has taken and what would have happened in the environment if agent was not there.

Often, the agents are required to choose their decisions based on the goal. When the goal information is combined with the past actions, it can choose better actions. It also has to check for optimization of the path, for finding the most optimal cost-effective solution to reach the destination.

We can see that the simple reflex agent is very rational as achieving the goal is relatively a simple task and there are no other agents influencing the environment. But in case of the automated taxi, it is almost impossible to achieve high rationality. It is a trade-off between many factors as there is no limit to the possible situations which can arise.

Therefore, Complex metrics have an advantage of Predictability, Inference, Automation, Training. On the other hand, they are effort intensive, less optimization, Dependency on other agents means more time consuming.

- b) No, a goal-based agent might not always perform better. The Goal-based agent lacks mainly with respect to efficiency. The main issue is that goal-based agents just work like binary switches as to whether the result is satisfactory or not. Example, in case of automated vehicle, there may be multiple routes which might even be more optimal. In such scenarios goal-based agents will not do complete justice. They compute all possible set of actions even before choosing to take any action and this would be very time consuming.

In case of a simple reflex agent, the task is pretty straightforward. We just follow a pre-defined set of actions. Even though this has its limitations and complexities, under optimal situations this agent would do more justice in reaching its destination as there is no preferences involved.

Let us say the Performance metric is to reach from Point A to B and the agent can move in all directions.

Pseudo code for goal-based agent is as follows:

Function goal-based-agent([location]) returns an action

    If goal achieved:

        Do nothing

    Else if:

        Plan all possible actions

        Choose best action

        Return action

Pseudo code for simple agent is as follows:

Function simple-agent([location]) returns an action

    If not at B:

        Move towards B

    Else if:

        Do nothing

As seen above, the goal based agent seems to be less efficient in comparison to the simple agent. This just hints towards path planning and search functions in AI.