

Exercise Solution
Chapter two : “INTELLIGENT AGENTS”

Name : Md Mynuddin

Roll : ASH1825007M

Year 3, Term 2

Question 2.1) Suppose that the performance measure is concerned with just the first T time steps of the environment and ignores everything thereafter. Show that a rational agent's action may depend not just on the state of the environment but also on the time step it has reached.

Answer 2.1: A rational agent's actions vary in regards to the environment. An action may or may not affect the environment, but if it does then we need to update the environment. One action may also lead to different paths, so we need to know what the final result will be in regards to every action, not just up until T time steps. An agent can only know what to do based on what it knows, if it stops after T time steps then there is a chance it will have not reached its goal, unless the goal in question has been designed around T time steps.

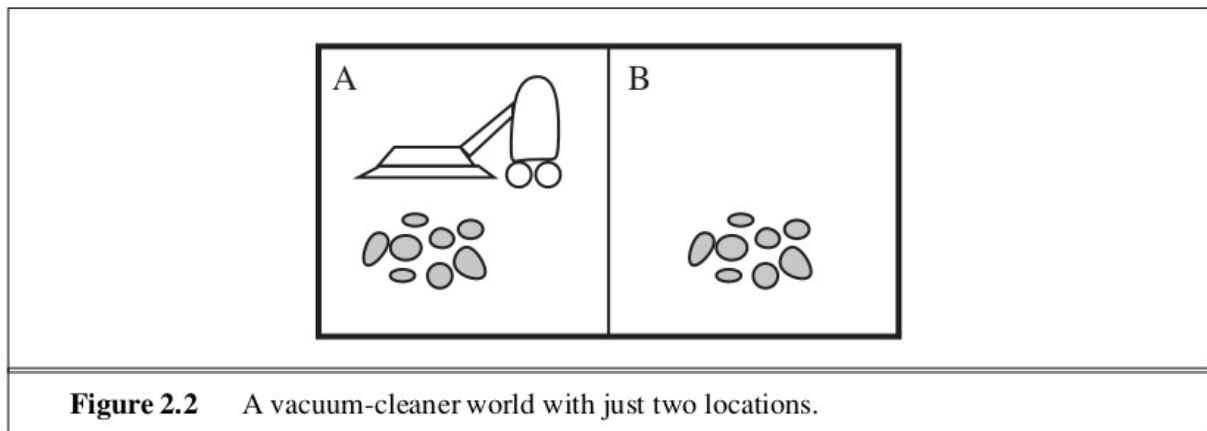
Question 2.2) Let us examine the rationality of various vacuum-cleaner agent functions.

- A)** Show that the simple vacuum-cleaner agent function described in Figure 2.3 is indeed rational under the assumptions listed on page 38.
- B)** Describe a rational agent function for the case in which each movement costs one point. Does the corresponding agent program require internal state?
- C)** Discuss possible agent designs for the cases in which clean squares can become dirty and the geography of the environment is unknown. Does it make sense for the agent to learn from its experience in these cases? If so, what should it learn? If not, why not?

Answer 2.2: The assumptions on page 38 state that:

- 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, and Suck.
- The agent correctly perceives its location and whether that location contains dirt.



Percept sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [A, Clean]	Right
[A, Clean], [A, Dirty]	Suck
⋮	⋮
[A, Clean], [A, Clean], [A, Clean]	Right
[A, Clean], [A, Clean], [A, Dirty]	Suck
⋮	⋮

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

A) The simple vacuum cleaner agent is indeed rational under these assumptions because it accounts for all variables. The map is known, there is a sensor for clean or dirty, what to do in that space, and the agent will never go out of bounds.

B) If each move costs one point then an internal state is required to keep track of starting points (if any) and the subtraction of those points. Unless of course the score starts at 0 and goes negative for each move. But if cleaning a space awards one point, then an optimal goal would be to try and get a score of 0 or higher.

C) If clean spaces can become dirty again then obviously a vacuum should clean it again. But the vacuum should not be constantly running, this would waste electricity. The vacuum should map its surroundings and objects so it

knows that the environment looks like for each additional time. By learning how large the area is it can determine how many times a day it must run to keep every space clean. Upon starting up it should assume every space is dirty and start a path around to clean them all, making sure to get every space. The more the vacuum does this the more it will know its surroundings and how often to clean.

Question 2.3) For each of the following assertions, say whether it is true or false and support your answer with examples or counterexamples where appropriate.

Answer 2.3 :

	True/False	Support
An agent that senses only partial information about the state cannot be perfectly rational.	False	The vacuuming cleaning agent is rational, despite it having no information about the other square.
There exist task environments in which no pure reflex agent can behave rationally.	True	A reflex agent reacts to the current percept, but if memory is needed then a pure reflex agent might behave irrationally.
There exists a task environment in which every agent is rational.	True	If every action results in the same result.
The input to an agent program is the same as the input to the agent function.	False	The agent program and function are different entities. The program contains history percepts and the function is only the current percept.
Every agent function is implementable by some program/machine combination.	False	A reflex agent will only act to certain variables.
Suppose an agent selects its action uniformly at random from the set of possible actions. There exists a deterministic task environment in which this agent is rational.	True	Eventually one of the random selections will be the correct one.
It is possible for a given agent to be perfectly rational in two distinct task environments.	True	Only if those environments are similar in a way.
Every agent is rational in an unobservable environment.	False	A GPS mapping system that doesn't record its findings.
A perfectly rational poker-playing agent never loses.	False	There is always a chance that its opponent will be dealt better cards.

Question 2.4) For each of the following activities, give a PEAS description of the task environment and characterize it in terms of the properties listed in Section 2.3.2.

- A) Playing soccer.
- B) Exploring the subsurface oceans of Titan.
- C) Shopping for used AI books on the Internet.
- D) Playing a tennis match.
- E) Practicing tennis against a wall.
- F) Performing a high jump.
- G) Knitting a sweater.
- H) Bidding on an item at an auction.

Answer 2.4:

Agent	Performance Measure	Environment	Actuators	Sensors
Playing Soccer	Score, Injuries, Team work.	Players, Referees, Field, Crowd, Goals, Ball.	Strength, Stamina, Coordination.	Eyes, Ears, Mouth, Ears, Touch.
Exploring Titan	Underwater mobility, Safety, Data, Navigation.	Shuttle, Rover, Atmosphere, Surface, Ocean.	Communication, Sustainability, Reliability.	Camera, GPS, Temperature, Pressure.
AI Book Shopping	Prices, Ease of site, Shipping time.	Websites, Internet, PC.	Correct Information, User.	Pictures, Information, Eyes.
Playing Tennis	Scoring, Stamina, Team work, Strategy.	Players, Referees, Crowd, Net, Court, Ball.	Strength, Stamina, Coordination.	Eyes, Skill, Footwork.
Practicing Tennis	Stamina, Lowering missed balls.	Player, Wall, Racket, Ball.	Stance, Racket Placement, Speed.	Eyes, Skill, Footwork.
High Jump	Form, Height, Landing.	Height bar, Padded Mat, Judge, Field.	Speed, Form, Leg Strength, Flexibility.	Eyes, Touch.
Knitting a Sweater	Correct Dimension, Reducing mistakes.	Yarn, Needles, Instructions, Room.	Speed, Yarn type, Sweater size, Precision.	Eyes, Hands.
Auction Bidding	Winning, Paying lowest price.	Opponents, Item, Auctioneer.	Budget, Item Value, Eagerness.	Eyes, Ears, Mouth, Knowledge of item.

Question 2.5) Define in your own words the following terms: agent, agent function, agent program, rationality, autonomy, reflex agent, model-based agent, goal-based agent, utility-based agent, learning agent.

Answer 2.5 :

Agent- A system with at least some form of intelligence.

Agent Function- What an agent is supposed to do, its purpose.

Agent Program- An internal absolute implementation of code.

Rationality- What the agent knows about the environment and a self judgement on how it performed.

Autonomy- The ability to act on its own. Knowing where it is, what it has to do, etc.

Reflex Agent- Responding to percepts in the environment.

Model Based Agent- Has knowledge of the workings of the world.

Goal Based Agent- Has knowledge of the goal and decides what actions to take in order to reach it.

Utility Based Agent- Determines the best way to reach the goal.

Learning Agent- Analyzes information to make improvements.

Question 2.6) This exercise explores the differences between agent functions and agent programs.

A) Can there be more than one agent program that implements a given agent function? Give an example, or show why one is not possible.

B) Are there agent functions that cannot be implemented by any agent program?

C) Given a fixed machine architecture, does each agent program implement exactly one agent function?

D) Given an architecture with n bits of storage, how many different possible agent programs are there?

E) Suppose we keep the agent program fixed but speed up the machine by a factor of two. Does that change the agent function?

Answer 2.6 :

A) Yes, there can be more than one agent program implementing an agent function. As stated above, the function is the purpose and the program is the code for its implementation. If a function has multiple options then there must be more than one program.

B) There exist agent functions that cannot be implemented by any agent programs. For example, if an agent function was to count to find the square root of a negative number. There is no way to solve that.

C) Yes, each agent program will implement exactly one agent function. For example, a precept has multiple reactions, but each reaction is different according to the situation it's in.

D) There are 2^n possible agent programs.

E) Speeding up the machine does not change the agent function because the environment is static.

Question 2.7) Write pseudocode agent programs for the goal-based and utility-based agents.

Answer 2.7 :

Goal-Based Pseudocode

set tickets_unsold to 50
set tickets_sold to 0
sell tickets for show
decrease tickets_unsold for each ticket_sold
increase tickets_sold for every decrease in tickets_unsold
stop when tickets_unsold equals zero

Utility-Based Pseudocode

set starting_location to (0,0)
set ending_location to (50,50)
create a fifty by fifty grid (even numbers are rows, odd numbers are columns)
set time to 0
randomly generate two numbers between 1 and 25 (the first is rows and the second is columns)
this determines how many roads contain traffic which increases time to ending_location
a normal block takes 1 minute to travel
a traffic block takes 5 minutes
start at (0,0) and arrive at (50,50) is the shortest amount of time
compute which path is the shortest and has the least amount of traffic.

Question 2.8) Implement a performance-measuring environment simulator for the vacuum-cleaner world depicted in Figure 2.2 and specified on page 38. Your implementation should be modular so that the sensors, actuators, and environment characteristics (size, shape, dirt placement, etc.) can be changed easily. (Note: for some choices of programming language and operating system there are already implementations in the online code repository).

Answer 2.8 :

Run the environment with this agent for all possible initial dirt configurations and agent locations. Record the performance score for each configuration and the overall average score.

Question 2.10) Consider a modified version of the vacuum environment in Exercise 2.8, in which the agent is penalized one point for each movement.

A) Can a simple reflex agent be perfectly rational for this environment?

Explain.

B) What about a reflex agent with state? Design such an agent.

C) How do your answers to a and b change if the agent's percepts give it the clean/dirty status of every square in the environment?

Answer 2.10:

A) A simple reflex agent cannot be perfectly rational in this environment because the agent never stops and its score will continue downward. It also has no idea whether there are even any unclean spaces before moving.

B) A reflex agent with a state is possible, as long as it keep track of the environment, otherwise it will keep moving from space to space. But the reflex agent performs the same action in similar situations, so entering a dirty space and moving is fine, but after moving from a clean space it will continue to move forever. So as long as the agent has memory of squares and the environment it is possible to work. There needs to be a line of code that states "after all squares are clean, stop).

C) If the agent knows whether a square is dirty or clean it has the option to take no action which prevents the score from decreasing. The agent should only clean dirty squares and if it has to travel to a dirty space, it should take the shortest route.