Middle East Technical University　　Department of Computer Engineering

CENG 462

Introduction to Artificial Intelligence

Fall '2015-2016

Programming Homework 4

Due date: 15 January 2016, Friday, 23:55

You Are Only Hope

Congratulations for discovering the secret teleportation machine located in the secret places of this world. Now, you have the ability to open a gateway to a parallel world ruled by Skynet.

As you unlocked and started the machine, the majestic gateway is opened. Now the question is; *what kind of world awaits* ?

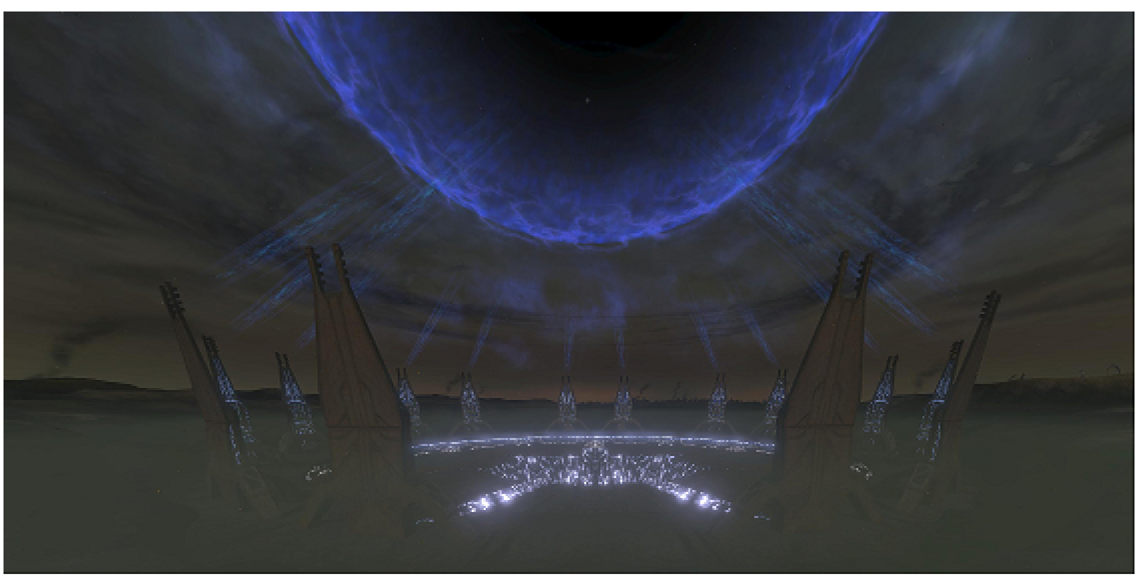

Figure 1: Teleport

As you guys entered the gateway, you found yourselves in a fairly dark room with transparent pipes on the wall having blue, green and purple liquid inside them, lighting the room. You noticed that there is a high-tech door, "NO MAN or KINGS, ONLY MECHA" written on top of it. As you approached to the door, two little minions, spider-like machines, come front and open the door. As the door opens, you see that the room (you are in) floats in a worm-hole like environment and there are teleportation pads ahead.

Once you try to step ahead to the pad in front, the pad buzzed you with electricity (remember the sign). Hopefully, the spider has a map to the environment and, of course, you know how to hack a mecha. Thus, you decided to hack one of the spider minions to send to the one of rooms, seen in the far distant. The idea is to pass one of the spiders to one of rooms ahead, so that the spiders can communicate and open portal for you to pass directly.

Figure 2: Program!!!

However, you sense that this is not the end of the story and there is more than meets the eye. You hack into the brain of the spider and gather some further details. After you pass to the other room you will reach the gate of my universe leading you to the environment where Skynet resides. You learn that first tele-pads work deterministically, and the tele-pads in my universe work in a probabilistic manner.

Here are the key points for the problem in figure below :

- Two different environments, i.e. the universes, are presented as a directed graph.
- The state transitions in your world is deterministic whereas in mine is probabilistic.
- The transitions between a universe is always one-way i.e. from your universe to mine and probabilistic. No turning back!!
- In this graph there will be 4 different types of nodes:

  Regular nodes in the bridge world (round nodes)

  Regular nodes in my world (star nodes)

  Teleportation nodes, the rooms i.e. bridges between the universes (square nodes)

  Vortex nodes, similar to black holes, they will suck your robot, robot shouldn't move to these nodes (cross nodes)

  Goal node, Skynet

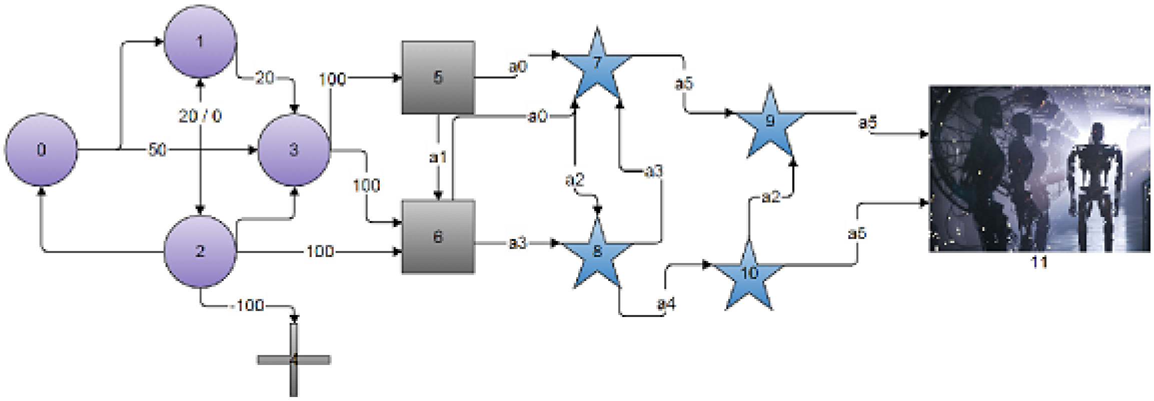

Figure 3: For round nodes the rewards are shown, for my plane the actions are shown. Note that they are stochastic, the probability distributions and rewards for the actions provided in the example file

Your task is to implement the Q**-learning** algorithm for the deterministic case and **Value Iteration** algorithm for the probabilistic case (and upload the code to the spider) :

- **Q-Learning**: Implement this algorithm for your universe;

  Initialize with all 0s.

  Goal nodes in here are the rooms. Do not consider further.

  Episodes will be given in the interactive session

- **Value Iteration**: Implement this algorithm for my universe;

  Starting nodes are the bridge nodes. Do not consider your universe.

  Initialize your value table with all 0s

  Goal node is Skynet

1 INPUT

Input will be given from file named *hw4.inp*, an example is provided at the end of this homework:

- The first line represents the type of nodes in the system ordered by their ID starting from 0: (R)ound ; (V)ortex ; r(O)om ; (S)tar ; (G)oal
- The second line represents the learning rate for the Q-learning algorithm and discount factor to be used in both of the algorithms
- Third line represents the total number of transitions for the deterministic plane, which the lines have the form :

  $< node1\_id > < node2\_id > < reward >$
- Next, the number of actions is given
- Then, for each node in my plane (except Skynet) the admissible set of actions are given in the form :

  $< node_i > < action\_id_0 >< action\_id_1 > \ldots < action\_id_n >$
- After those lines, the effects of each action is represented in the form :

  action : $< action\_id_i >$

  $< node\_id_i >$

  $< reward >$

  $< node\_id_0 > < transition\_probability\_as\_percentile >$

  $< node\_id_1 > < transition\_probability\_as\_percentile >$

  ...

  $

  $< node\_id_j >$

  $< reward >$

  $< node\_id_0 > < transition\_probability\_as\_percentile >$

  $< node\_id_1 > < transition\_probability\_as\_percentile >$

  ...

  $

  #

  action : $< action\_id_j >$

  ... *(same here)*

2 INTERACTIVE SESSION & OUTPUT

After building the world, your code will expect the following string of ids of nodes for episodes for Q-learning algorithm, examples are:
$hw4.py >$ 0 1 2 1 3 5
$hw4.py >$ 0 2 0 1 2 3 6

After each episode, your program should print the Q-Table in the form of a matrix, for impossible transitions (like from 0 to 5 above), print _. For telling your program that the Q-learning is finished:
$hw4.py >$ $

After terminating the Q-learning with the character $, your program should print two lists ; where the first list represents the value table while the list represents your policy for the first iteration. After, your code repeatedly will wait for character c for continuing iteration (one step) then print the lists again, and wait for the character $ for termination.

$hw4.py > c$
$hw4.py >$ $

For the Q-table and the value table (V table) , for an environment where 0 has neighbours 1,2, 1 has neighbours 0,2, 2 has neighbours 1,3. 3 has no neighbours, then an example representation of the table can be;

_ 10 10 _
20 _ 10 _
_ -5 _ 50
_ _ _ _

For the policy list you should use the following format;

- $< node\_id_0 > \quad < node\_id_1 >, < node\_id_2 >, \ldots, < node\_id_n >$

where the first node represents the name of the node the latter represents the best states to go from the $< node\_id >$. For example, for an environment where the policy states that: go to 1 or 2 from state 0, go to 2 from 1, go to 3 from 2, then the output should be;

0 $1, 2$
1 2
2 3

Your code should be in Python.

Example *hw4.inp* :

RRRRVOOSSSSG
0.5 0.5
11
0 1 0
0 3 50
1 2 20
1 3 20
2 0 0
2 1 0
2 3 0
2 4 -100
2 6 100
3 5 100
3 6 100
6
5 0 1
6 0 3
7 2 5
8 3 4
9 5
10 2 5
action : 0
5
0
7 80
8 20
$
6
0
7 80
8 20
$
#
action : 1
5
0
6 100
$
#
action : 2
7
-10
8 50
5 30
6 20
$
10
20
9 80

8 20
$
#
action : 3
6
-10
8 90
5 10
$
8
100
7 50
5 50
$
#
action : 4
8
0
10 80
7 20
$
#
action : 5
7
20
9 90
5 10
$
9
200
11 80
10 20
$
10
200
11 80
8 20
$
#
E

3 Regulations

1. Save the World.
2. Your code should be in Python.
3. **Late Submission: Not Allowed**
4. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating will be punished according to the university regulations.
5. **Newsgroup:** You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.
6. **Evaluation:** The .py file will be checked for plagiarism automatically using "black-box" technique and manually by assistants, so make sure to obey the specifications.

4 Submission

Submission will be done via COW. You code file should be named "hw4.py". Do not send archives.

Note: You cannot submit any other files. Don't forget to make sure your .py file is successfully compiled in Inek machines.