B 551: Elements of AI
Fall 2016 - Assignment 5

Ritesh Tawde/rtawde@iu.edu

Q.2)

The approach used in solving mdp is by using Policy Iteration.
Policy iteration is implemented using following two approaches:
1) Policy evaluation:

Given a policy π_i, calculate the utility of each state if π_i were to be executed with some default policy

2) Policy improvement:

Calculate a new policy π_{i+1}, using one-step look-ahead using the following equation:
Max($\sum_{s^‘ \in S} P(s^‘|s, \pi[s])(R(s, a, s^‘) + \gamma V[s^‘])$, $(\sum_{s^‘ \in S} P(s^‘|s, a)(R(s, a, s^‘) + \gamma V[s^‘])$
where, S = state space
s = current state
$s^‘$ = next state
pi[s] = policy for state s
γ = discount factor
V[s] = value function(utility) for state s
R(s) = reward for being in state $s^‘$ from state s with action a
and update the policy for each state if satisfied by the above formula
Discount factor is selected high (between 0 and 1) to encourage for future rewards and not taking greedy approach.

Above approach continuously improves the policy until no best policy is found for state and action.

Q.3

a) State space for the full observable grid:
For a grid world with 4x4 space, state space consists of
S = [(0,0),(1,0),(2,0),(3,0),(0,1),(1,1),(2,1),(3,1),(0,2),(1,2),(2,2),(3,2),(0,3),(1,3),(2,3),(3,3)]
along with wumpus-dead and has-arrow states
State space is excluding walls if any

b) Set of actions :
A = ['do nothing','left','right','up','down','shoot left','shoot right','shoot up','shoot down']

c)Transition function:
- If next state is in wall locations, state is not changed with probability of 0 going in wall locations
- If moved in the intended direction, movement occurs with probability of 0.9 in the intended direction and 0.1 elsewhere
- If shot in a particular direction for wumpus and the wumpus is in next location, it returns 1 as the highest probability of going in next state from the current state
- If found gold, return 'do nothing' action with probability of being in gold location as 1

Following table summarizes the transition function:

Current State(s)	Actions(a)	Next state(s`)	Probability(p)
x,y	up or down or left or right	x+1,y or x-1,y or x,y+1 or x,y-1 (in wall locations)	0.0
x,y	do nothing	x,y (same state)	1.0
x,y	do nothing	x,y(gold)	1.0
x,y	up	x,y+1	0.9
x,y	up	x-1,y or x+1,y or x,y-1	0.1/3
x,y	down	x,y-1	0.9
x,y	down	x,y+1 or x+1,y or x-1,y	0.1/3
x,y	left	x-1,y	0.9
x,y	left	x,y+1 or x+1,y or x,y-1	0.1/3
x,y	right	x+1,y	0.9
x,y	right	x,y+1 or x,y-1 or x-1,y	0.1/3
x,y	shoot up	x,y<wumpus-location[y] in wumpus-location	1.0
x,y	shoot down	x,y<y¡wumpus-location[y] in wumpus-location	1.0
x,y	shoot left	x<wumpus-location[x],y in wumpus-location	1.0
x,y	shoot right	x<wumpus-location[x],y in wumpus-location	1.0

d) Reward function: Reward is -100 for pit or wumpus, +100 for gold and -1 elsewhere.

```
1) Reward for pit location = R(pit) = -100
2) Reward for wumpus location if wumpus not dead = R(wump-loc|not wumpus-dead)=-100
3) Reward for gold location = R(gold) = 100
4) Reward for being in any other state = R(other) = -1
```

Q. 4)

Given map of height h and width w, state space formalization consists of
$w * h$ + wumpus-dead + has-arrow - walls if any

REFERENCES:

`http://artint.info/html/ArtInt_228.html`
`Reinforcement Learning, An Introduction by Sutton and Barto, second edition(draft)`