EL2805 Reinforcement Learning
Computer Lab 1

Jingxuan Mao
001214-9068
jmao@kth.se

Yuqi Zheng
990122-1243
yuqizh@kth.se

January 10, 2023

1 Problem 1: The Maze and Random Minotaur

1.1 Assumptions

In this lab, we assume that the Minotaur is able to enter all walls.

1.2 Basic Maze

(a) Formulate the problem as an MDP.

- State space

  $\mathcal{S} = \{((p_x, p_y), (m_x, m_y)) : (p_x, p_y) \text{ and } (m_x, m_y) \text{ are reachable}\}$

  where (p_x, p_y) and (m_x, m_y) refer to the position of the player and the Minotaur, respectively.

- Action space

  $\mathcal{A} = \{left, right, up, down, stay\}$

- Reward

  Step: If at state s, taking action a, leads the player to some other position in the maze that is not the exit nor a wall nor an obstacle, then $r(s, a) = -1$

  Goal: If at state s, taking action a, leads the player to the exit B, then $r(s, a) = 0$

  Eaten: If at state s, taking action a, leads the player to be eaten by the Minotaur, then $r(p_x = m_x, p_y = m_y|s, a) = -100$

  Impossible: If at state s, taking action a, leads the player to a wall or an obstacle then $r(s, a) = -200$

- Transition probabilities

  If at state s taking action a does not lead to a wall or an obstacle or the player being eaten by the Minotaur but to another state s', then $\mathbb{P}[s' = ((p'_x, p'_y), (m'_x, m'_y))|s, a] = \frac{1}{n_m}$;

  If at state s taking action a leads to a wall or an obstacle or the player being eaten by the Minotaur, the player will remain in his position, while the Minotaur can move randomly, then $\mathbb{P}[s' = ((p_x, p_y), (m'_x, m'_y))|s, a] = \frac{1}{n_m}$, where n_m is the number of all the possible states the Minotaur can reach at the next position under all possible actions. In the lab we just randomly sample one state of the Minotaur since it's movement is stochastic.

(b) Model modified problem as an MDP.
When the player and the Minotaur do not move simultaneously, but in alternative rounds, the player would be able to observe the current position of the Minotaur, which means the player can avoid approaching the Minotaur more easily. Thus, it will be more difficult for the Minotaur to catch the player. Compared with the original MDP, the state of the modified MDP contains an additional element representing the the player's or the Minotaur's turn.

1.3 Dynamic Programming

(c) Find an optimal policy.
As shown in Figure 1, under a policy that maximizes the probability of leaving the maze alive, the path of the player is:

(0, 0), (1, 0), (1, 1), (2, 1)
(3, 1), (4, 1), (4, 2), (4, 3)
(4, 4), (4, 5), (4, 6), (4, 7)
(5, 7), (6, 7), (6, 6), (6, 5)

the path of the Minotaur is:

(6, 5), (4, 5), (4, 4), (4, 5)
(6, 5), (6, 6), (6, 5), (6, 3)
(4, 3), (3, 3), (3, 1), (2, 1)
(2, 0), (3, 0), (4, 0), (5, 0)

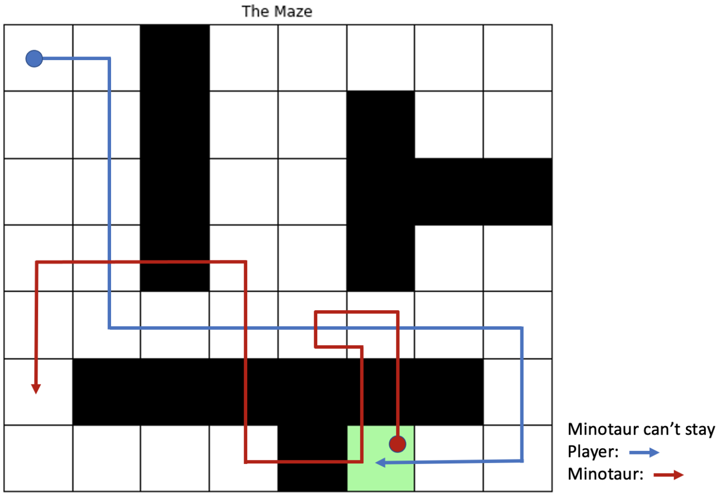

Figure 1: A policy that maximizes the probability of leaving the maze alive.

(d) Plot the probabilities of exiting the maze alive.
For $T = 1, 2, ..., 30$, the probabilities of exiting the maze alive is shown is Figure.

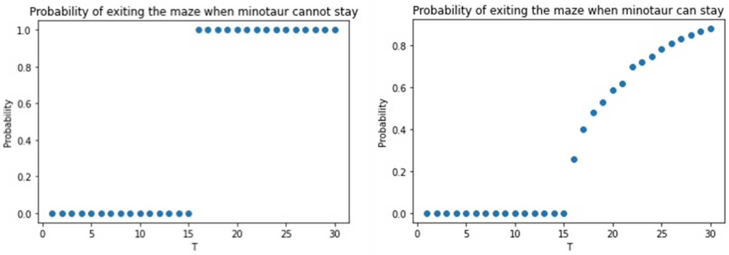

Figure 2: Probability of exiting the maze alive.

We can see that when the Minotaur is not allowed to stand still, from T = 16, the probability of exiting the maze successfully would be 1. If the Minotaur is allowed to stay at the same position, the probability of the player exiting the maze alive would increase with T. However, the player will have a lower chance to exit the maze alive, since the Minotaur could block the path to the exit.

1.4 Value Iteration

(e) Modify the problem with poison.
The life of the player is geometrically distributed with mean 30, which means the discount factor $\lambda = 1 - \frac{1}{30} = \frac{29}{30}$. In addition, we set ϵ to be 1e-5.
(f) We simulate the path under this policy for 10000 times and find the probability of exiting the maze alive is 0.604 and 0.502 when the Minotaur is and is not allowed to stand still, respectively.

1.5 Additional Questions

(g) Answer theoretical questions.
1) On-policy learning means learning the value of the policy under which the data are generated.
Off-policy learning refers to learning the best policy regardless of the generation process of the data.
2) Assume the decreasing step sizes (α_t) satisfy

$$\sum_t \alpha_t = \infty$$

and

$$\sum_t \alpha_t < \infty$$

For Q-learning, further assume that the behavior policy π_b can visit every (state, action) pairs infinitely often.
(h) Modify the MDP.
We divide the problem into two stages: from A to C, and then from C to B. For each process, we have a similar MDP to the previous one, where $\lambda = 1 - \frac{1}{50} = \frac{49}{50}$. However, in this case, we include another element in the state to indicate whether the player gets the key or not, and the reward is 0 only when the player gets the key. The transition probabilities become $\mathbb{P}[s'|s,a] = \frac{1}{0.35+0.65(n_m-1)}$ and $\mathbb{P}[s'|s,a] = \frac{1}{0.35+0.65(n_m-1)}$, respectively.