**ECE 457C – Reinforcement Learning**

**Assignment 2**

**Professor Mark Crowley**

**Matthew Erxleben, 20889980**

**June 26th, 2024**

# Part 2: Implement Four Core TD Methods

## Q2: Definition of Design of Algorithms

**States:**

Each State represents the RL Agents position in the Grid World, on the 10x10 matrix. These states have coordinates that represent its position, in a list [X, Y].

**Actions:**

Each action can be either up, down, left, or right. This reflects how the Agent will move throughout the Grid World. When the Agent decides one of the four actions, its respective state will change to move to where ever it is moving in the Grid World. This is all based on its previously defined [X, Y] coordinates.

**Dynamics:**

The state changes when the Agent decides an action, based on its policy. The Agent will move to the state after applying their action. The Agent receives a reward based on its action and its resulting state. In the Grid World, there is the starting position (denoted by the red square), the Goal position (denoted by the yellow circle), walls (denoted by the black squares), and pits (denoted by the blue squares). Each of these positions have respective attributes and rewards to them. Every blank position gives a reward of -0.1, and the agent moves into that position if their action directs them there. If the agent tries to walk into a wall, they stay in their current position and receive a reward of -0.3. If the agent trues to walk into a pit, they receive a reward of -10 and the episode is terminated. If the Agent reaches the goal state, they receive a reward of 1 and it ends the episode.

**Notation:**

S: the current state

$S'$: the next state

A: the current action

$A'$: the next action

R: after taking action A in state S, the agent recieves the reward R

$\alpha$: the learning rate

$\gamma$: the discount factor

$\pi(a \mid S')$: the probabilty of taking action a, given you are in state $S'$

**Bellman Updates:**

Each algorithm uses their respective Bellman Updates to update their Q-values. This is based on the expected future rewards, and the current reward received.

**SARSA:**

SARSA is on-policy, and uses this Bellman update:

$$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$$

**Q-Learning:**

Q-Learning is off-policy, and uses this Bellman update:

$$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma max_a Q(S',a) - Q(S,A)]$$

**Expected SARSA:**

$$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \sum_a \pi(a \mid S') Q(S',a) - Q(S,A)]$$

**Double Q-Learning:**

With probability 0.5:

$$Q_1(S,A) \leftarrow Q_1(S,A) + \alpha\left[R + \gamma Q_2\left(S', argmax_a Q_1(S',a)\right) - Q_1(S,A)\right]$$

else:

$$Q_2(S,A) \leftarrow Q_2(S,A) + \alpha\left[R + \gamma Q_1\left(S', argmax_a Q_2(S',a)\right) - Q_2(S,A)\right]$$