Question 1:

Given the TD(0) update rule:

$$V(S_t) = V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

Where:

- $V(S_t)$ is the current value estimate of state S_t.
- α is the learning rate.
- R_{t+1} is the reward received after transitioning from state S_t.
- γ is the discount factor.
- $V(S_{t+1})$ is the value estimate of the next state S_{t+1}.

Given:

- $V(E) = 2$
- $V(F) = 2$
- $\gamma = 0.5$
- $\alpha = 0.5$
- The sequence observed is: $E, a, 1, F, b$

Using the TD(0) update rule for state E:

$$V(E) = V(E) + \alpha[R_{t+1} + \gamma V(F) - V(E)]$$
$$V(E) = 2 + 0.5[1 + 0.5(2) - 2]$$
$$V(E) = 2 + 0.5[1 + 1 - 2]$$
$$V(E) = 2 + 0.5[0]$$
$$V(E) = 2$$

So, after the TD(0) update, the state-value $V(E)$ remains 2.

Question 2:

Given the SARSA update rule:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

Where:

- $Q(S_t, A_t)$ is the current action-value estimate for state S_t and action A_t.
- α is the learning rate.
- R_{t+1} is the reward received after taking action A_t in state S_t.
- γ is the discount factor.
- $Q(S_{t+1}, A_{t+1})$ is the action-value estimate of the next state S_{t+1} and next action A_{t+1}.

Given:

- $Q(E, a) = 2$
- $Q(F, a) = 0$
- $\gamma = 0.5$
- $\alpha = 0.5$
- The sequence observed is: $E, a, 1, F, a$

Using the SARSA update rule for action a in state E:

$$Q(E, a) = Q(E, a) + \alpha[R_{t+1} + \gamma Q(F, a) - Q(E, a)]$$
$$Q(E, a) = 2 + 0.5[1 + 0.5(0) - 2]$$
$$Q(E, a) = 2 + 0.5[1 - 2]$$
$$Q(E, a) = 2 + 0.5[-1]$$
$$Q(E, a) = 2 - 0.5$$
$$Q(E, a) = 1.5$$

So, after the SARSA update, the action-value $Q(E, a)$ becomes 1.5.

Question 3:

Given the Q-learning update rule:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t)]$$

Where:

- $Q(S_t, A_t)$ is the current action-value estimate for state S_t and action A_t.
- α is the learning rate.
- R_{t+1} is the reward received after taking action A_t in state S_t.
- γ is the discount factor.
- $\max_{a'} Q(S_{t+1}, a')$ is the maximum action-value estimate for the next state S_{t+1} over all possible actions a'.

Given:

- $Q(E, a) = 1$
- $Q(F, a) = 3$
- $Q(F, b) = 4$
- $\gamma = 0.5$
- $\alpha = 0.5$
- The sequence observed is: $x0 = E, u0 = a, r1 = 0, x1 = F$

Using the Q-learning update rule for action a in state E:

$$Q(E, a) = Q(E, a) + \alpha[r1 + \gamma \max_{a'} Q(F, a') - Q(E, a)]$$

Given that the maximum Q-value for state F is $Q(F, b) = 4$:

$$Q(E, a) = 1 + 0.5[0 + 0.5(4) - 1]$$
$$Q(E, a) = 1 + 0.5[2 - 1]$$
$$Q(E, a) = 1 + 0.5$$
$$Q(E, a) = 1.5$$

So, after the Q-learning update, the action-value $Q(E, a)$ becomes 1.5.

Question 4:

The correct answer is:

True

Explanation:
Monte-Carlo (MC) methods estimate the value function based on the average return from many episodes. As the number of episodes approaches infinity, the Law of Large Numbers ensures that the Monte-Carlo estimate will converge to the true expected value. Thus, for discrete (tabular) state spaces, given enough samples/episodes, the MC estimate of the state value function V is guaranteed to converge to the true value.

Question 5:

The correct answer is:

True

Explanation:
For discrete (tabular) state spaces, under certain conditions, the TD(0) estimate of the state value function V is guaranteed to converge to the true value. Specifically, if every state is visited infinitely often and the learning rate α meets the Robbins-Monro conditions (i.e., $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$), then the TD(0) estimate will converge to the true value function. This convergence property is one of the reasons why TD methods are popular in reinforcement learning.

Question 6:

The correct answer is:

True

Explanation:
TD/SARSA methods can indeed be used to estimate the action-value function $Q(x, u)$ in both episodic and continuing (non-episodic) environments. In continuing environments, the agent-environment interaction doesn't break into identifiable episodes but goes on continually. SARSA, being an on-policy TD control method, updates its estimates at each time step based on the current state, action, reward, next state, and next action. This incremental update mechanism makes it suitable for continuing tasks as well as episodic tasks.

Question 7:

The correct answer is:

False

Explanation:
Monte-Carlo (MC) methods estimate values based on complete episodes. They require the episode to terminate to calculate the return (cumulative reward) from a state or state-action pair until the end of the episode. In continuing (non-episodic) environments, where there is no terminal state and the interaction goes on indefinitely, it's not straightforward to apply traditional MC methods because there's no clear endpoint to calculate the return. Thus, MC methods are typically not used for estimating action-value functions in continuing tasks.

Question 8:

The correct answer is:

False

Explanation:
Tabular methods represent the value of each state or state-action pair individually in a table. As the state space grows, the table grows, requiring more memory. For large or continuous state spaces, tabular methods become impractical due to the sheer size of the table.

Function approximation methods, on the other hand, use a parameterized function to represent the value function or policy. This allows them to generalize across similar states and handle much larger or even continuous state spaces without explicitly storing a value for every possible state. Common function approximation techniques include neural networks, linear function approximators, and basis function methods.

Thus, function approximation methods are more scalable and can handle larger state spaces than tabular methods.

Question 9:

The correct answer is:

True

Explanation:
The optimal action value function, denoted as Q^*, represents the maximum expected return achievable from a given state after taking a specific action, following the optimal policy thereafter. For a given task/environment, there is a unique Q^* that represents the highest achievable expected return for each state-action pair. However, while Q^* is unique, the optimal policy π^* that achieves this optimal action value function may not be unique. There can be multiple optimal policies that achieve the same optimal action value function, especially in environments where multiple actions can lead to the same maximum expected reward in certain states.

Question 10:

The correct answer is:

True

Explanation:
Q-learning is an off-policy TD control algorithm that updates its estimates of the action-value function based on the maximum expected future rewards. Unlike Monte-Carlo methods, Q-learning does not rely on episodes to complete in order to update its estimates. Instead, it updates its Q-values incrementally at each time step. This property allows Q-learning to be applied in both episodic and continuing (non-episodic) environments. In continuing tasks, Q-learning can keep updating its Q-values as it interacts with the environment, making it suitable for such scenarios.