SARSA Algorithm for Grid World

Let's consider a simple grid world with 3x3 cells, where the agent can move up, down, left, or right. The grid has a start state S and a goal state G. The agent receives a reward of -1 for each step and a reward of +10 for reaching the goal state. The discount factor γ is set to 0.9.

Initialization:

- Initialize the action-value function $Q(s, a)$ arbitrarily for all s and a.
- Initialize the state S to the start state.
- Choose an action A using an exploration policy (e.g., epsilon-greedy).

Algorithm:

1. **Repeat** for each time step:
   - Take action A and observe the reward R and the next state S'.
   - Choose the next action A' using the same exploration policy.
   - Calculate the TD target:

$$\text{TD target} = R + \gamma \cdot Q(S', A')$$

   - Update the action-value function:

$$Q(S, A) \leftarrow Q(S, A) + \alpha \cdot (\text{TD target} - Q(S, A))$$

     where α is the learning rate.
   - Set S to S' and A to A'.

Final Optimal Policy:

- The final optimal policy π can be derived from the learned action-value function $Q(s, a)$.
- For each state s, choose the action a that maximizes $Q(s, a)$:

$$\pi(s) = \arg\max_a Q(s, a)$$

- The optimal policy π specifies the best action to take in each state to maximize the expected cumulative reward over time.