Reinforcement Learning
Exercise 5 - Solution

Jonathan Schnitzler - st166934
Eric Choquet - st160996

June 1, 2024

Task 1) - Random Walk

In the Random walk example from lecture 5 slide 12 the value is a prediction of the probability of terminating on the right side of the chain. Let us first recall the update rule for $TD(0)$:

$$V(S_t) \leftarrow V(S_t) + \alpha \left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right] \tag{1}$$

The only changes in value (V(that can occur after the first episode are at the ends since they are the only sates where a reward is generated. In our case only $V(A)$ is updated, the sampled Random walk terminated at the left. Then the equation above evaluates to

$$V(A) \leftarrow V(A) + 0.1 \left[0 + 0 - V(A)\right] = V(A) - 0.1V(A) = 0.5 - 0.1 \cdot 0.5 = 0.45 \tag{2}$$

Otherwise, also the state of another value would have been changed, e.g. if first the Random walk would have gone to the right, then $V(B)$ would have been updated as well.

Eric: The actions leading to the first reward cannot be traced back as the updates bring no changes due to the uniform initialisation of the values.

$$V(B) \leftarrow V(B) + 0.1 \left[0 + V(A) - V(B)\right] = V(B) - 0.1[0.5 - 0.5] = 0.5 \tag{3}$$

Task 2) - Sarsa and Q-learning on the Frozen Lake

a) - Sarsa The Sarsa algorithm is implemented in the accompanying python file. The algorithm is run for 10,000 episodes with a learning rate of $\alpha = 0.1$ and a discount factor of $\gamma = 0.9$. The results are shown in the following figures. One notices that the tiles with value 0 are either holes or the goal. The algorithm has learned to avoid the holes and reach the goal in the lower right corner.

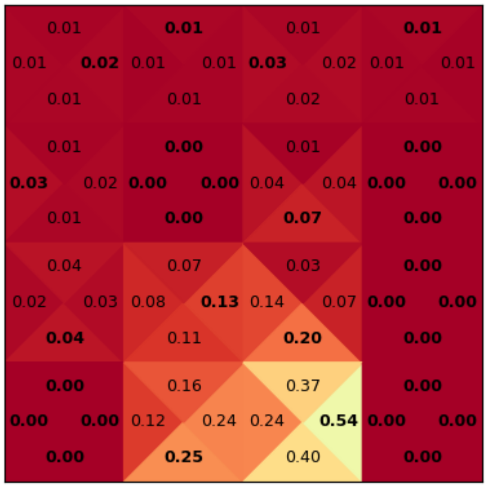


Figure 1: Action-value matrix Q for Sarsa on the Frozen Lake

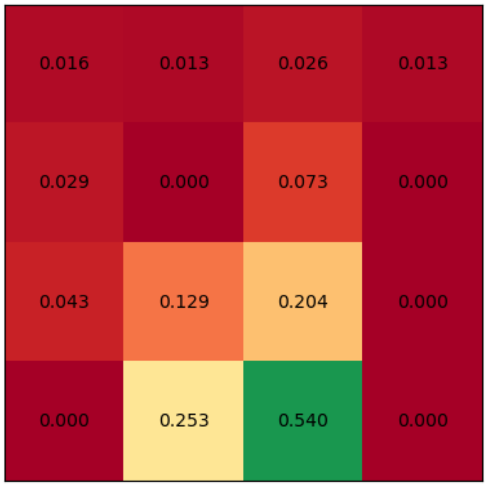


Figure 2: Value matrix V for sarsa on the Frozen Lake

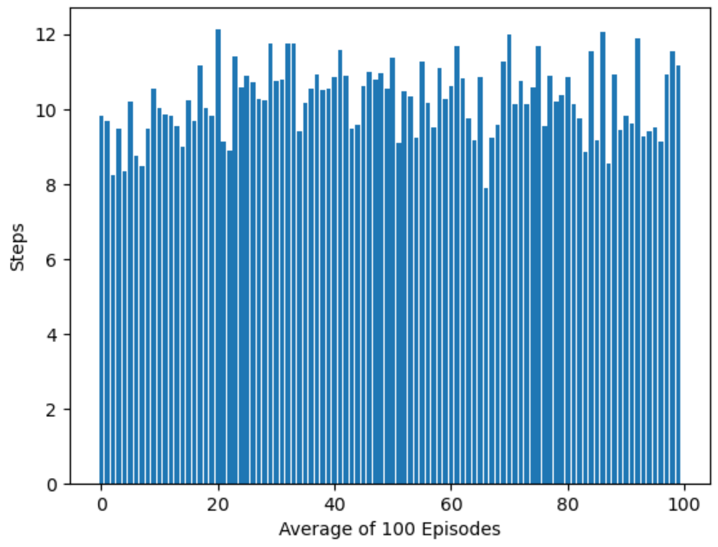


Figure 3: Average episode length as training continues

b) Q-learning The Q-learning algorithm is implemented in the accompanying python file. The algorithm is run for 10,000 episodes with a learning rate of $\alpha = 0.1$ and a discount factor of $\gamma = 0.9$. The results are shown in the following figures. The main difference is that for the update step the maximum over all possible actions is taken, instead of the action that is actually taken. Therefore the performance during training is worse, then the performance for optimal policy.

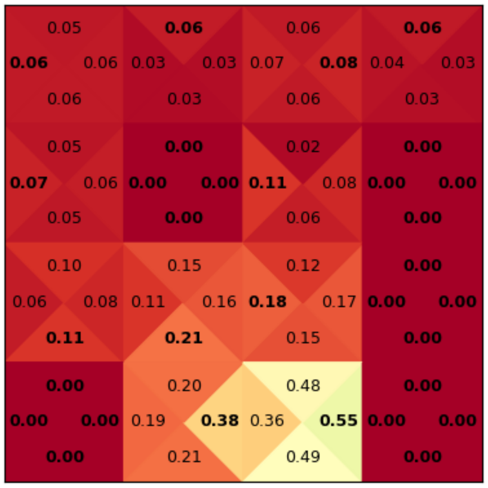


Figure 4: Action-value matrix Q for Q-learning on the Frozen Lake

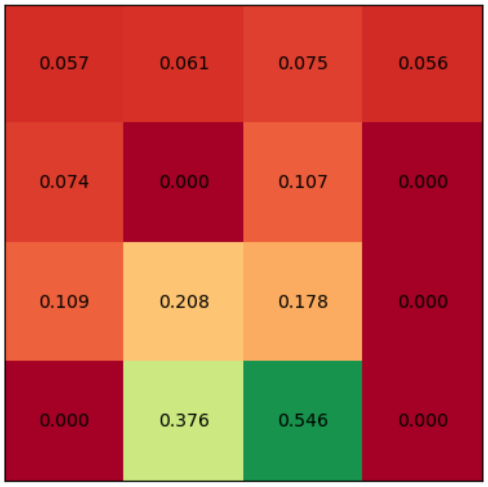


Figure 5: Value matrix V for Q-learning on the Frozen Lake

c) Deterministic policy The same plots but now evaluated for the deterministic policy of the frozen lake. Q learning outperforms Sarsa in this case.

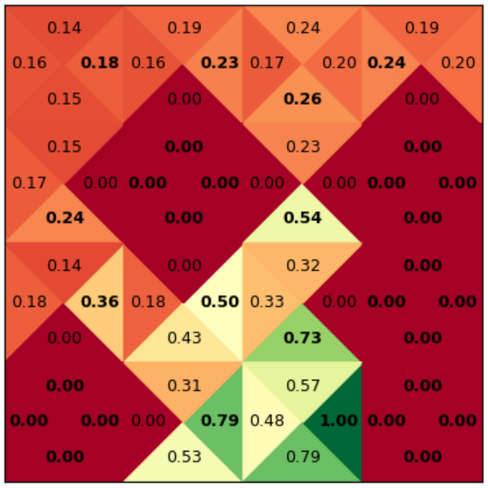


Figure 6: Deterministic Action-value matrix Q for Sarsa on the Frozen Lake

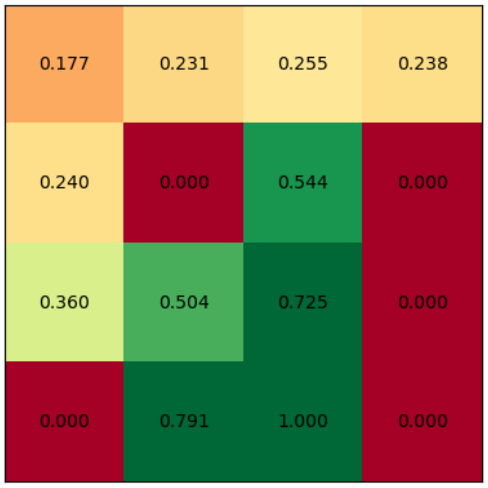


Figure 7: Deterministic Value matrix V for sarsa on the Frozen Lake

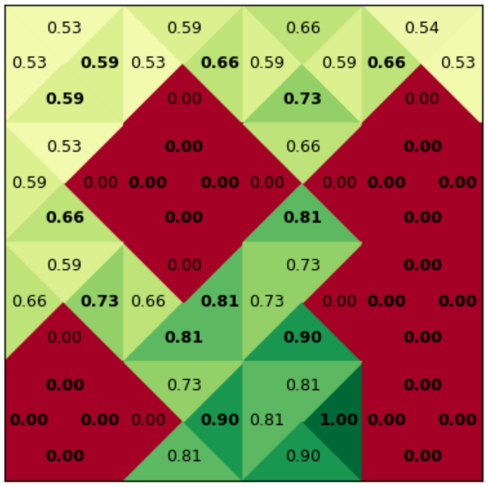


Figure 8: Deterministic Action-value matrix Q for Q-learning on the Frozen Lake

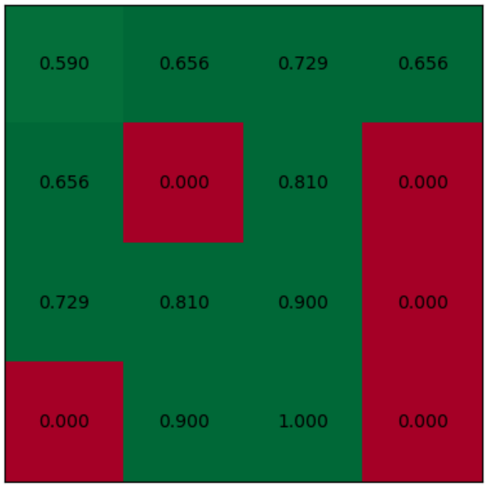


Figure 9: Deterministic Value matrix V for Q-learning on the Frozen Lake

d) - Larger grid For the larger 8x8 grid 1,000,000 episodes were run for both algorithms, as otherwise the convergence was not reached. The results for sarsa are shown.

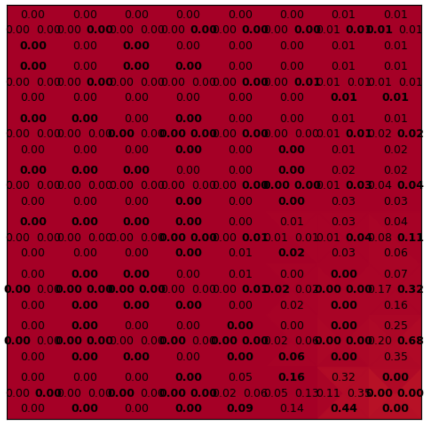
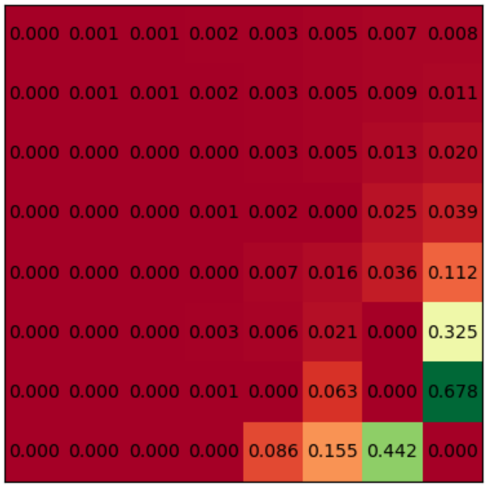


Figure 10: Action-value matrix Q for Sarsa on the Frozen Lake 8x8 grid



0.000	0.001	0.001	0.002	0.003	0.005	0.007	0.008
0.000	0.001	0.001	0.002	0.003	0.005	0.009	0.011
0.000	0.000	0.000	0.000	0.003	0.005	0.013	0.020
0.000	0.000	0.000	0.001	0.002	0.000	0.025	0.039
0.000	0.000	0.000	0.000	0.007	0.016	0.036	0.112
0.000	0.000	0.000	0.003	0.006	0.021	0.000	0.325
0.000	0.000	0.000	0.001	0.000	0.063	0.000	0.678
0.000	0.000	0.000	0.000	0.086	0.155	0.442	0.000

Figure 11: Value matrix V for sarsa on the Frozen Lake 8x8 grid