

CS670 - Reinforcement Learning

Puddle World Programming Assignment

Gabriel Dulac-Arnold <gabe@squirrelsoup.net> (CS09F004)
Johannes H. Jensen <johannj@stud.ntnu.no> (CS09F005)

March 4, 2017

1 Q-Learning and Sarsa

Q-Learning and Sarsa were both trained using a constant exploring probability $\epsilon = 0.1$ and learning rate $\alpha = 0.1$ over 10 000 episodes, averaged over 50 runs.

1.1 Goal A

Goal A gives rapid convergence and pretty clear policies. We notice that the policy generated by SARSA seems a bit less chaotic than Q-learning, but we attribute this difference to chance.

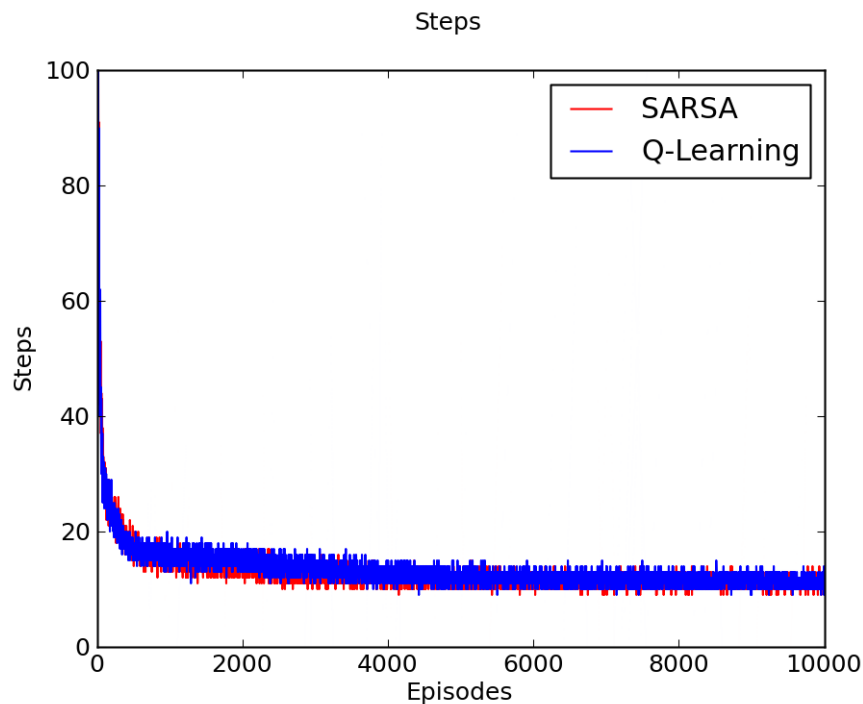


Figure 1: Average number of steps to reach goal

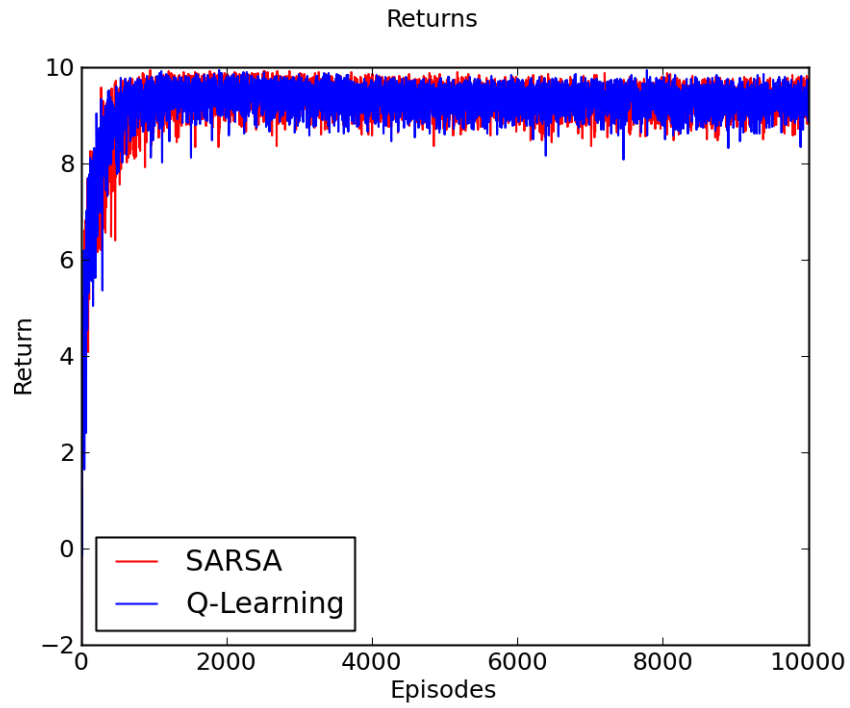


Figure 2: Average reward per episode

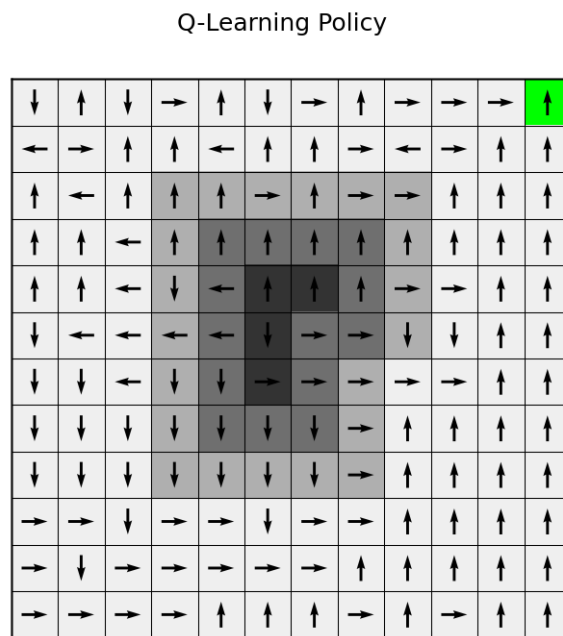


Figure 3: Policy learned by Q-Learning

SARSA Policy

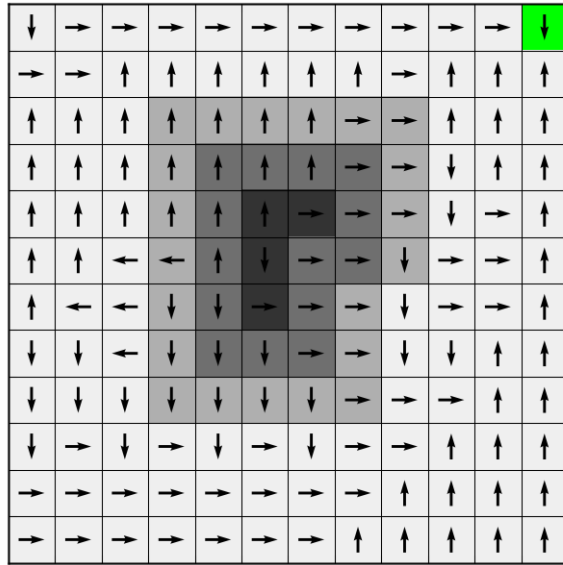


Figure 4: Policy learned by Sarsa

1.2 Goal B

Here we see similar results as those with goal A. There is a very minor peak in average reward, that seems to decrease slightly as the episodes go on. We have no clear hypothesis for why this is hapenning, although it could have something to do with our choice of constant ϵ for the ϵ -greedy search.

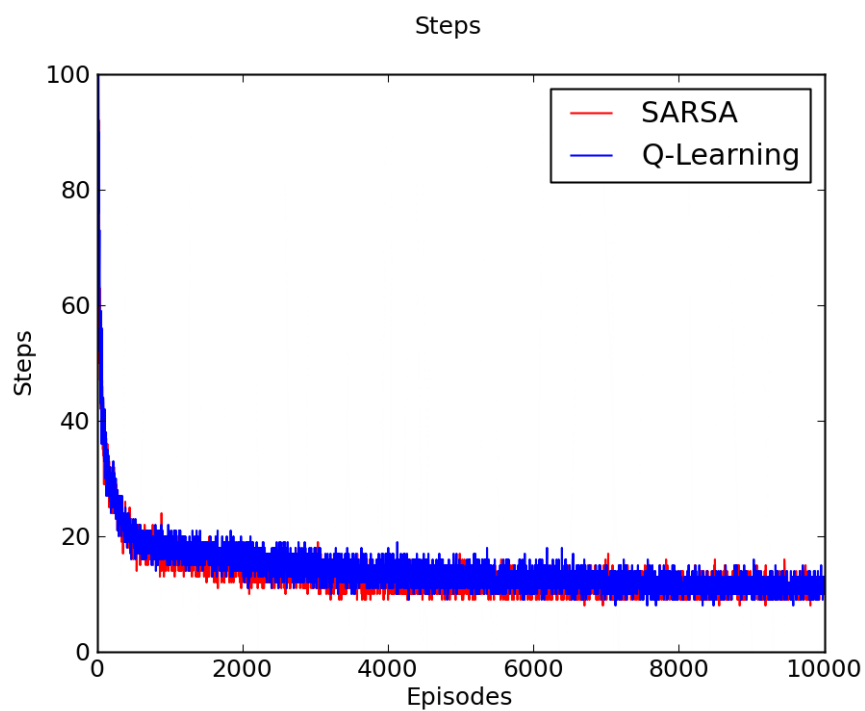


Figure 5: Average number of steps to reach goal

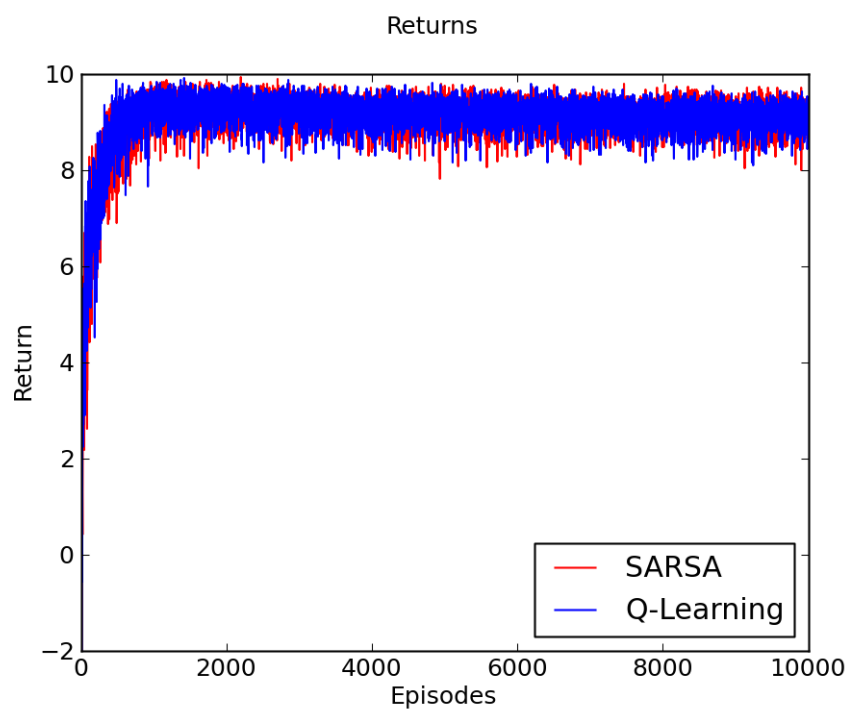


Figure 6: Average reward per episode

Q-Learning Policy

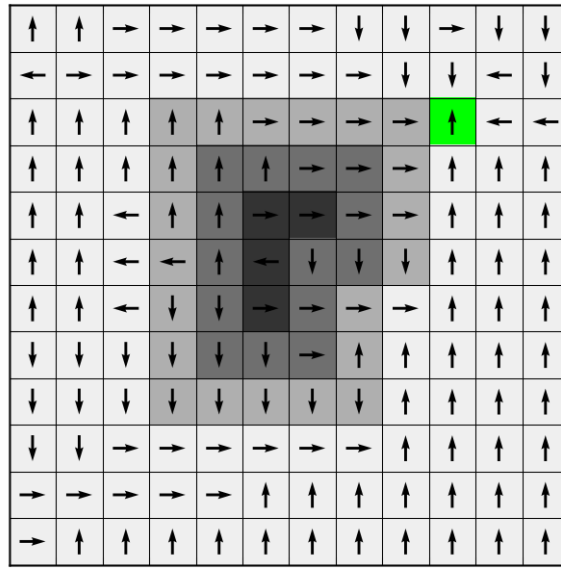


Figure 7: Policy learned by Q-Learning

SARSA Policy

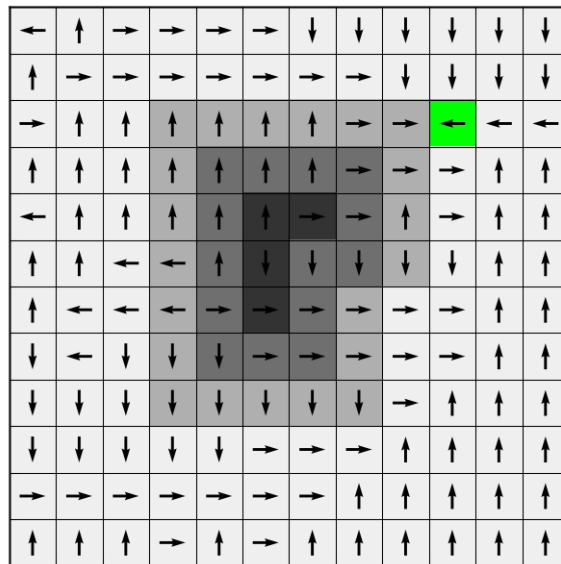


Figure 8: Policy learned by Sarsa

1.3 Goal C

Both SARSA and Q-Learning continue to perform well even with this more difficult goal. The only big difference with this run is that returns peak after about 1000 episodes and then decrease quite visibly. Once again there is no clear cause, and our hypotheses are the same as mentioned in goal B.

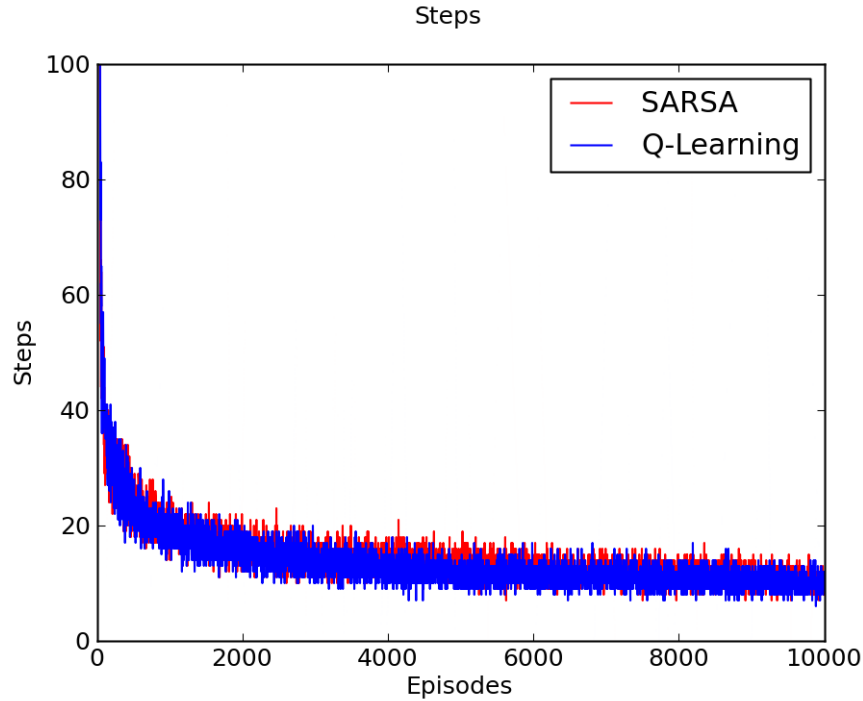


Figure 9: Average number of steps to reach goal

2 Policy Gradient

Policy Gradient was run on an environment with deterministic actions using learning rate $\alpha = 0.005$ and initial temperature $\tau = 500$, which decreased to $\tau = 1$ by the 300th episode. The graphs are averaged over 30 runs. This method only converged for Goal A.

We notice here that we seem to attain an optimal number of steps after 2000 episodes, and then the number of steps bumps up twice. This would be worrying except that it seems the longer policies give better returns, as we can see in figure 14. Only after 4500 episodes or so are we consistently getting a reward of 8.

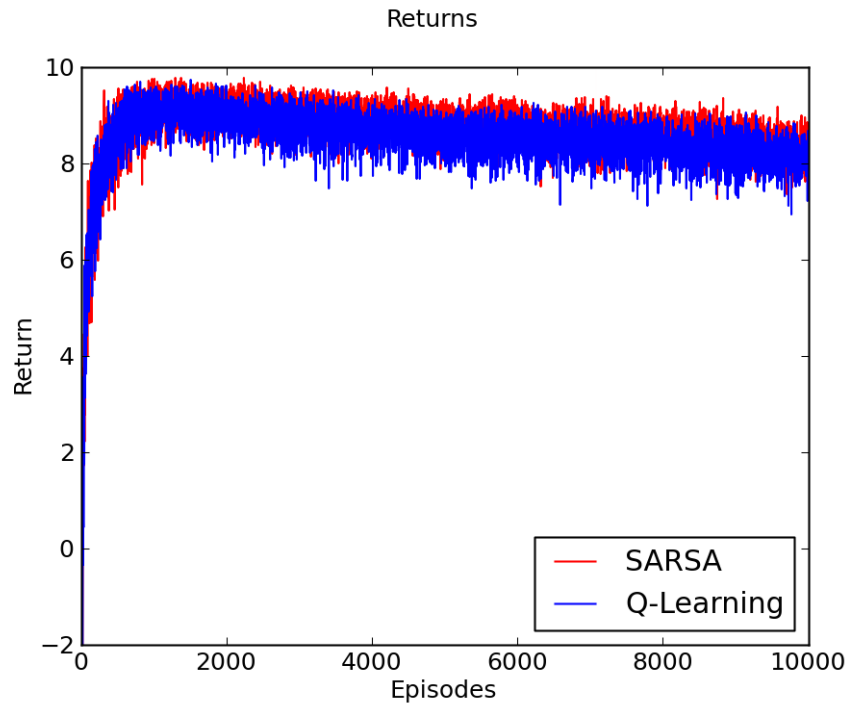


Figure 10: Average reward per episode

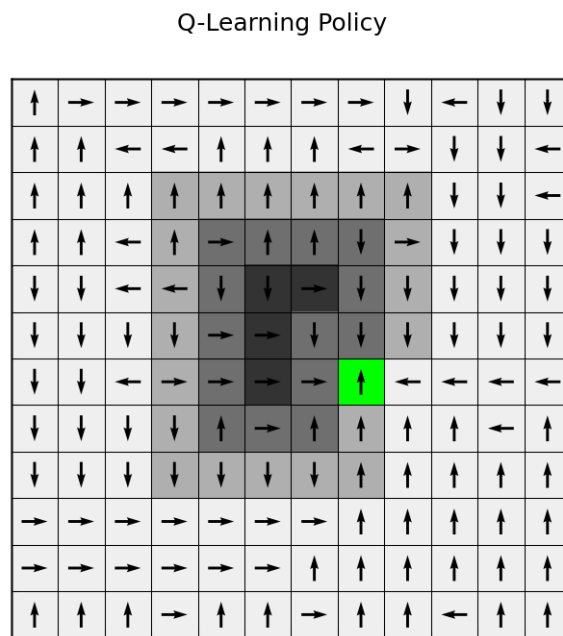


Figure 11: Policy learned by Q-Learning

SARSA Policy

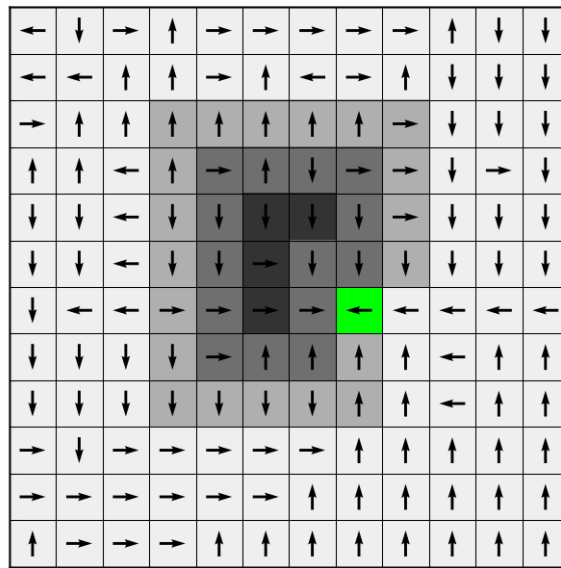


Figure 12: Policy learned by Sarsa

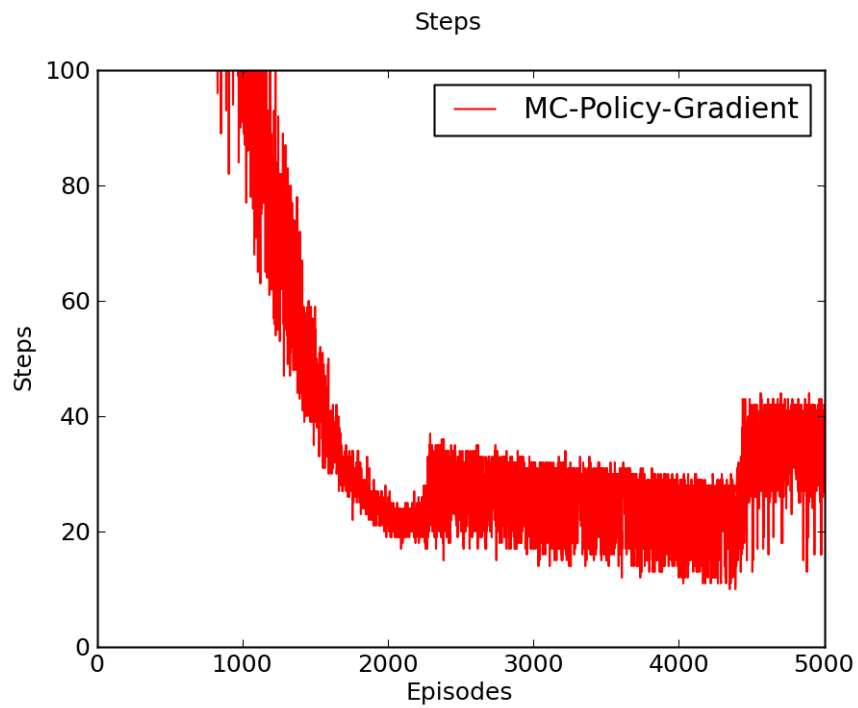


Figure 13: Average number of steps to reach goal

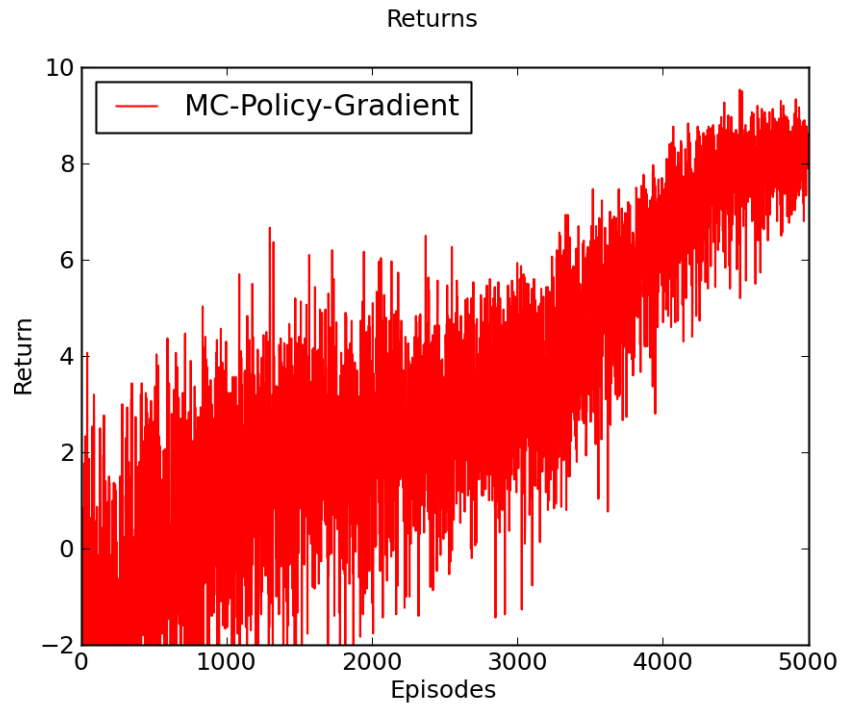


Figure 14: Average reward per episode

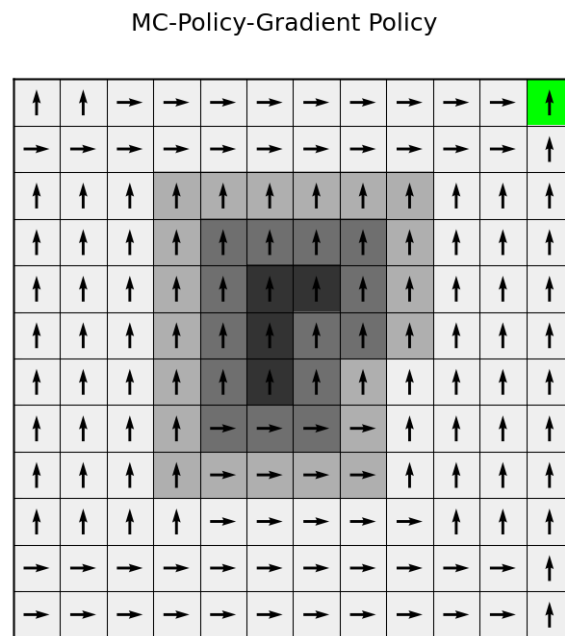


Figure 15: Policy learned by MC Policy Gradient

The uniformity of this policy is characteristic of a policy gradient approach. Even inside the puddle, areas which we would have most likely explored less, we can see that the policy is sensible. This is carried over from the experience of the agent in the zones around the puddle and is one of the clear advantages of policy gradient.