RL - Reinforcement Learning

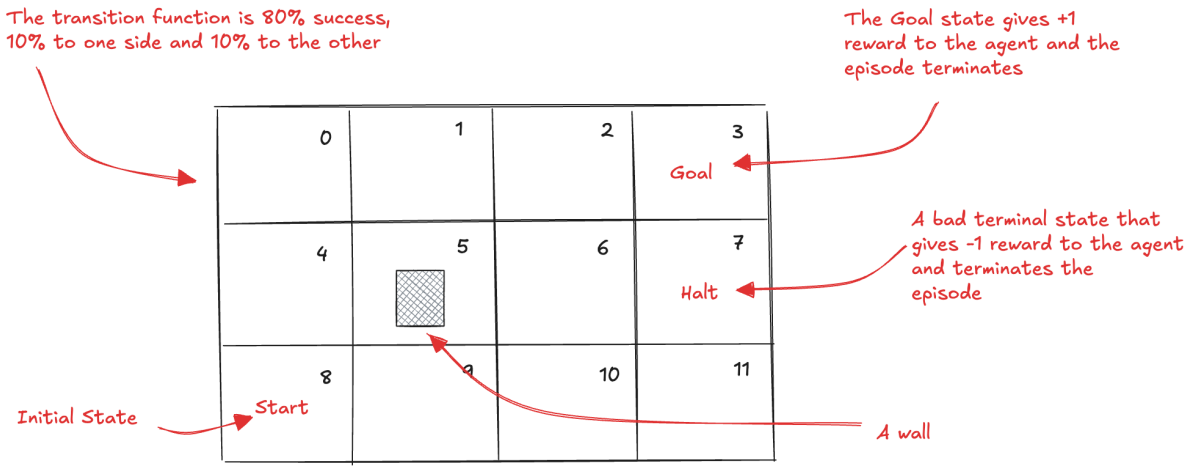
Exercises on Policy Evaluation

Prof. Riccardo Berta

2025.10.03

1 Gridworld

Create the environment from the Russell and Norvig's book on AI: **the Gridworld**. This environment is a 3 x 4 grid world in which the agent starts at the bottom-left corner, and it has to reach the top-right corner. There is a hole south of the goal and a wall near the start. The transition function has a 20% noise; that is, 80% the action succeeds, and 20% it fails uniformly at random in orthogonal directions. The reward function is a –0.04 living penalty, a +1 for landing on the goal, and a –1 for landing on the hole.

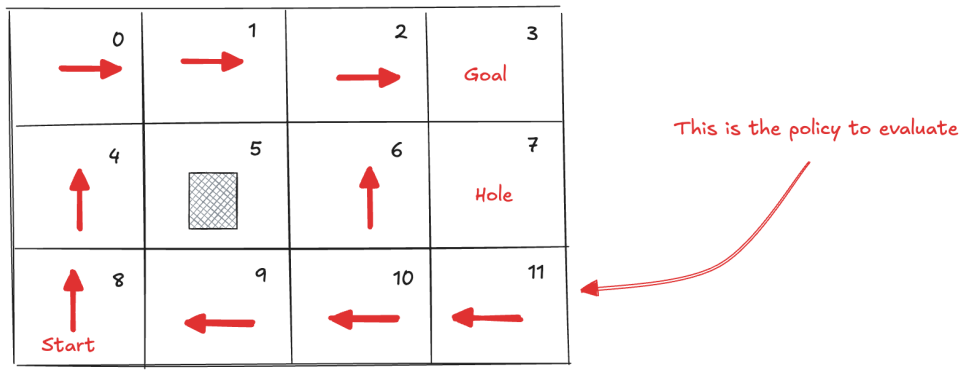


1 - Create the environment Gridworld as described above.

```
# YOUR CODE HERE

# The environment should be a class with the following methods:
# `__init__(self)`: initialize the environment
# `reset(self)`: reset the environment to the initial state
# `step(self, action)`: take an action in the environment.
# The action should be an integer between 0 and 3, where 0 means left,
# 1 means down, 2 means right, and 3 means up.
# The method should return a tuple (state, reward, done),
# where state is the new state, reward is the reward obtained, and done is
# a boolean indicating whether the episode is finished.
```

2 - Write the following policy for this environment




```
# YOUR CODE HERE

# The policy should be a function that takes an integer between 0 and 11
# (the state) and returns an integer between 0 and 3 (the action).
```

3 - Show the policy by printing the selected action for each state:

```
# YOUR CODE HERE

# The function should print the policy in a human-readable format.
# For example, the output could be:
# state: 0 → action: RIGHT
# state: 1 → action: RIGHT
# etc ...
```

4 - Now, evaluate this policy using TD(λ)

```
# YOUR CODE HERE

# you need to decay alpha
```

```
# YOUR CODE HERE

# you need to decay the discount
```

```
# YOUR CODE HERE

# you need the td_lambda implementation
```

```
# YOUR CODE HERE

# run the algorithm and print the value function
```

5 - Plot the estimated value for state 8, 6 and 2 for all episodes

```
# YOUR CODE HERE

# plot the value function over episodes for state 8, 6 and 2
```