Stellenbosch UNIVERSITY IYUNIVESITHI UNIVERSITEIT

Department of Mathematical Sciences
Division of Applied Mathematics

Introduction to Reinforcement Learning

Coursework 2

- Optional for after the course.
- Comes with two demos:
  - Gymnasium and the minigrid environment
  - Creating videos from gymnasium simulations

Tabular Q-learning

Implement an agent solving the `MiniGrid-Empty-8x8-v0` problem in `gymnasium` using Q-learning (see the two demos). The agent must implement ε-greedy exploration, with ε decaying during training. The hyperparameters are the learning rate α, the discount factor γ, and the minimum and maximum values, ε_{min} and ε_{max}, of ε.

- Train a policy for the agent for a maximum of 2000 episodes using Q-learning. Plot the reward and number of steps to completion as a function of the training steps.
- Use a python dictionary to store the Q-value function with a hash function to generate a unique key for each state of the agent. You might want to use the `hashlib` package for that.
- Evaluate the performance of your final trained policy by calculating the reward and number of steps you get for just one episode. Use your policy in a greedy way, not ε-greedy way.
- Perform hyperparameter tuning to find optimal training values resulting in the highest average reward and completion rate.

Hugo Touchette
Version: 05/08/2026 16:46