COMPSCI 687 Homework 3 - Fall 2025

Due **November 1**, 11:55pm Eastern Time

1 Instructions

- This homework assignment consists of a written portion and a programming portion.
- While you may discuss problems with your peers (e.g., to discuss high-level approaches), you must answer the questions on your own. In your submission, do explicitly list all students with whom you discussed this assignment.
- Submissions must be typed. You must use LaTeX. Handwritten and scanned submissions will not be accepted.
- The assignment should be submitted on Gradescope as a PDF with marked answers via the Gradescope interface. The source code should be submitted via the Gradescope programming assignment as a .zip file. Include with your source code instructions for how to run your code. You **must** use Python 3 for your homework code.
- You may not use any reinforcement learning or machine learning specific libraries in your code, e.g., TensorFlow, PyTorch, or scikit-learn. You *may* use libraries like numpy and matplotlib, though.
- The automated system will not accept assignments after 11:55pm on November 1.
- The .tex file for this homework can be found on Canvas under "*Homework #3 - Supporting Files*".

Part Two: Programming (55 Points Total)

In this question, you will implement the Value Iteration algorithm and use it to find the optimal value function and the optimal policy for a domain called the Cat-vs-Monsters domain. The domain you should implement is described below.

Notice that you may <u>not</u> use existing RL code for this problem. You must implement the agent and environment entirely on your own and from scratch.

- **States**: This problem consists of a 5×5 environment where each state $s = (r, c)$ describes the current coordinates/location of a cat. In particular, $r \in [0, 4]$ is the current row where the cat is located, and $c \in [0, 4]$ is the current column where the cat is located. Refer to Figure 1 for an example. In this figure, the topmost row is row zero, and the leftmost column is column zero—i.e., *State1* corresponds to $s = (0, 0)$ and *State14* corresponds to $s = (3, 1)$.
- **Actions**: There are four actions: AttemptUp (AU), AttemptDown (AD), AttemptLeft (AL), AttemptRight (AR).
- **Dynamics**: This is a *stochastic* MDP:
  - With 70% probability, the cat moves in the specified direction.
  - With 12% probability, the cat gets confused and moves to the right with respect to the intended direction.
  - With 12% probability, the cat gets confused and moves to the left with respect to the intended direction.
  - With 6% probability, the cat gets sleepy and decides not to move.
  - The environment is surrounded by walls. If the cat hits a wall, it gets scared and does not move.
  - There are four *Forbidden Furniture* locations in this environment: one in $(2, 1)$, one in $(2, 2)$, one in $(2, 3)$, and one $(3, 2)$. If the cat touches a Forbidden Furniture, it gets paralyzed and remains in its current state. The cat cannot go on the furniture.
  - There are two *Monsters*: one in $(0, 3)$ and one in $(4, 1)$.
  - There is a *Food state* located at $(4, 4)$.
- **Rewards**: The reward is always -0.05, except when transitioning to (entering) the Food state, in which case the reward is 10; or when transitioning to (entering) a state containing a Monster, in which case the reward is -8. Notice that to model this type of reward function, you will need to use a reward function in the form $R(S_t, A_t, S_{t+1})$ instead of $R(S_t, A_t)$. This requires a small modification to the Value Iteration update equation: $v_{i+1}(s) := \max_{a \in \mathcal{A}} \sum_{s'} p(s, a, s') \Big(R\big(s, a, s'\big) + \gamma v_i(s') \Big)$.
- **Terminal State**: The Food state is terminal. Any actions executed in this state transition to s_∞ with reward 0.
- **Discount**: Unless otherwise specified, $\gamma = 0.925$.
- **Initial State**: The cat deterministically wakes up at the beginning of each episode on its bed (i.e., $S_0 = (0, 0)$).

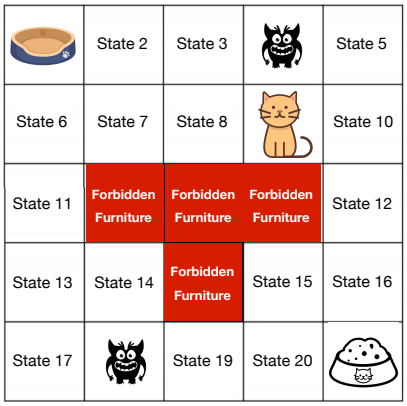

Figure 1: The Cat-vs-Monsters domain.