598RL: HW #1

T. Bretl M. West

(due by 10AM on Thursday, September 6)

1 A pursuit-evasion game

Consider the 4×4 grid with boundary that is shown in Figure 1. You control the good robot (blue circle). At each time step, the good robot can stay in the same place or can move one cell north, south, east, or west. The good robot cannot move through the boundary (i.e., out of the grid)—if it tries, the result is to stay in the same place. The evil robot (red triangle) has a mind of its own. At each time step, the evil robot either stays in the same place (20% chance) or moves one cell in a direction chosen at random (20% north, south, east, or west). Like the good robot, the evil robot cannot move through the boundary—if it tries, it stays in the same place. The good robot would like to spend as much time at the goal (green square) as possible. This goal has a fixed location. If the good robot ends a time step in the same cell as goal, it receives a reward of +1. If the good robot instead ends a time step in the same cell as the evil robot, the game is over. Each time the game is played, the good robot, the bad robot, and the goal are each chosen uniformly at random (so, it is possible for a game to have length zero). The good robot knows its own location as well as the location of both the bad robot and the goal at each time step.

- Formulate this game as an MDP.
- Write python code that. . .
- Visualize an example game. . .

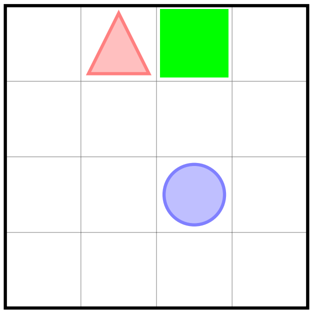

Figure 1: The playing field for a game of pursuit-evasion.