3547 Assignment 3

This assignment builds on your Beeline Agent. The new agent will keep the same strategy for planning the route home after grabbing the gold but will have a new added superpower: a strategy for searching the grid for the gold as safely as it can. This will require it to make inferences about which squares definitely don't have a pit or Wumpus, or for those that might, what the probability is so it can take calculated risks.

Instructions

1. Create a copy of the BeelineAgent called ProbAgent (because it will use probabilistic reasoning) or make ProbAgent a subclass of Beeline Agent if you prefer. You'll need to add additional belief state variables to track the history of locations of Stench and Breeze percepts. You'll need these to make inferences about the probability of there being a Wumpus or pit at locations the agent hasn't visited yet, which will enable it to perform much better than even a Logical Agent. You'll also want to keep track of whether you heard a Scream or not.
2. Use the Python Pomegranate library to build two causal graphs, one for the relationship between pit locations and breeze locations, and one for the relationship between Wumpus locations and stench locations. The first graph should encode how the locations of the pits cause breezes at other grid locations. Initialize the graphs with the prior probabilities of there being a pit or Wumpus at each location.
3. The agent should use the percepts it receives at each step to update the state variables mentioned above, which will change the probabilities of there being pits and a Wumpus at other locations.
4. Use the probabilities which can be queried from the two models to decide how best to search the grid for the gold. You *don't* want your agent to (a) *unnecessarily* visit locations it's already been (b) take *unnecessary* risks (c) give up without attempting to kill the Wumpus if it is likely to be beneficial, (d) waste its arrow *unnecessarily*, or (e) to give up unless the next move is more than 50%[1] likely to result in its death (in which case it should proceed as quickly as it can to (1,1) and Climb). You may find that a modified version of your route planner from the BeelineAgent can help speed up the search part of the game as well. Your agent will need to take risks and occasionally die to play its long-run best, but they should be calculated risks and only when there are no risk-free paths known to be available. Note also that the arrow isn't just a weapon; it can also be used as a probe for information about the Wumpus' location.
5. What is your agent's total score after exactly 1,000 games?

[1] You won't know what the real probability of success is, jus the probability of dying on the next move. If the chance of dying on the next move is 50-50, the probability of successfully completing the maze will a be a bit lower than that, so under the rules of the game the best strategy in that situation would be to abort the mission. If we increase the payoff for success from +1,000 to something larger or add a penalty for quitting empty-handed, the probabilistic agent will clearly outperform a logical agent by accepting the occasional low-probability death in return for a greater long-run reward. A logical agent can't distinguish low and high risk moves, just safe and unsafe ones.