Problem Description

Visual example of 2x2 grid with 1 goal and 1 obstacle Github

Input

The input to the problem, i.e., the domain, is an n by n grid of squares, a set of goals that must be accomplished in order, and a set of obstacles that block movement.

The combination of position and goal states (accomplished/not accomplished) make up the "state".

Output

The output to the problem is a set of four values for each state. These values indicate affinity for each direction. They are called Q-values.

Evaluation

Currently, PRISM outputs a set of probabilities corresponding to:

1. The probability of the goals being achieved individually.
2. The probability of each sequence of goals being achieved (i.e, goal 1; goal 1 -> 2; goal 1, 2 -> 3, …).
3. The probability of hitting a moving obstacle.

This assumes that the starting point is (0, 0). This score is weighted arbitrarily to produce an "ltl score". For now, this is used as a simple evaluation metric.

Transition Probabilities

PRISM uses softmax with a temperature τ to turn Q-values (affinity for a direction) into transition probabilities.

$$P(a|s) = \frac{\exp(Q(s,a)/\tau)}{\sum_{a'} \exp(Q(s,a')/\tau)}$$

Basic LLM testing

I have refactored the code to work with any number of obstacles and any number of goals. When testing the LLM in a 2x2 grid with 1 obstacle and 1 goal, the LLM achieves the best path given the current evaluation metrics.

Next steps

1. Set transition probability to 1 in the direction of the highest Q-value and 0 everywhere else.
2. Make the LLM only plan along a single path, not a set of Q-values for each state, and modify PRISM accordingly.
3. Allow the LLM to plan in multiple directions, not only a single direction.
4. Expand to 3x3 grid
5. Add another goal
6. Add more static obstacles
7. Introduce moving obstacles

Possible improvements

- Calculate the average probability of reaching a goal from any arbitrary position.
- Use PRISM's reward model to calculate a more interesting result (e.g., lower reward for high probability of hitting obstacles, etc.). Currently, the PRISM output does not really interpret the probability of hitting obstacles well at all.