Nathan Thom

Problem Set [0]

CS 482

9/18/2018

**Question 1:**

**Part 1: Formulate this problem. How large is the state space?**

**a state space is the set of all possible configurations of a system.**

States:

Each state is determined by two things. First, the robot’s position in the maze. Second, by the direction that the robot is currently facing. The state can be shown in the following form: (X, Y, D). X is the horizontal coordinate of the robot. Y is the vertical coordinate of the robot. Finally, D is the position that the robot is currently facing. D can be one of the directions in the following set of possible directions: {North, South, East, West}.

The agent can be in any space contained by the maze that is not a wall, let’s call this set P. This means that there are |P| \* 4 states that the robot can be in. In words, the cardinality of the set of possible spaces times the number of positions that the robot can be oriented in within each space.

Initial State:

The initial state of the robot is the space that is closest to the middle of the maze as possible. This space must be in the set of possible spaces. In an ideal maze this would be: (max(X)/2, max(Y)/2, N). In words, the maximum possible horizontal distance divided by 2, the maximum vertical distance divided by 2 , and oriented to face North.

Actions:

There are two possible actions for the robot: moving and turning. The robot can turn(D), where D is the direction that the robot should turn to face, and move(Z) where Z is the number of spaces to traverse.

Transition Model:

Beginning in the initial state, there are two transition functions for our two actions. First, (current\_state(X1,Y1,D1), turn(D)) = new\_state(X1,Y1,D2) and second (current\_state(X2,Y2,D1), move(Z)) = new\_state(X2,Y2,D2)

Goal Test:

For any valid maze there should be at least 1 goal state. That is there should be some position in the set of possible positions that is defined the goal position. The goal state is this position with the robot in any orientation. The goal test will check to see that the robot is in a goal position, if it is then the task is complete.

Path Cost:

Every move will cost 1 and every turn will cost 1.

The state space is |P| \* 4. The cardinality of the set of possible positions multiplied by 4.

**Part 2: In navigating a maze, the only place we need to turn is at the intersection of two or more corridors.**

States:

Now the state space has changed. The robot can either be in a corridor or at an intersection of two or more corridors. While in a corridor the robot cannot turn. This means that the state can be shown in the following form: (X,Y,D,C or I) where C denotes corridor and I denotes intersection. We still need to know where the robot is and which direction it is facing so that we know when we arrive at an intersection space as well as which direction we are already facing so that we know which direction to move.

Initial State:

The initial state stays mostly the same except to accommodate the addition of corridors in the state of the robot. The new initial state should take the form: (max(X)/2, max(Y)/2, N, C or I).

Actions:

There are three possible actions for the robot, now. Moving, turning, and checking for intersections of two or more corridors. The robot can turn(D) when at an intersection, where D is the direction that the robot should turn to face, move(Z) where Z is the number of spaces to traverse, and lastly check\_for\_I(X, Y) where X and Y indicate the current position of the robot.

Transition Model:

Beginning in the initial state, there are three transition functions for our three actions.

First (current\_state(X1,Y1,D1,I), turn(D)) = new\_state(X1,Y1,D2,I)

Second (current\_state(X1,Y1,D1,C or I), move(Z)) = (current\_position(X2, Y2), check\_for\_I(X2,Y2)) = new\_state(X2,Y2,D1,C or I)

Goal Test:

For any valid maze there should be at least 1 goal state. That is there should be some position in the set of possible positions that is defined the goal position. The goal state is this position with the robot in any orientation. The goal test will check to see that the robot is in a goal position, if it is then the task is complete.

Path Cost:

Every move will cost 2 because the move needs to be executed and then the check for an intersection needs to be executed. Every turn will cost 1.

The state space must now be updated. We have the set P of possible positions for the robot and now we also have the set T. The set T is a subset of set P and contains states in which it is possible for the robot to turn in, or in other words positions that are intersections of two or more corridors. This means that the state space is now 2(|P| - |T|) + 4|T|. Or in words 2 multiplied by the number of corridor spaces and 4 multiplied by the number of intersection spaces. Each corridor can be traversed in two directions and each intersection can be occupied while facing 4 directions.

**Part 3: From each point we can move in any of the 4 directions until we reach a turning point. Moving in any of the 4 directions is the only action that we need to worry about.**

States:

The possible states of the robot are simplified. We only need to know the robot’s position in the maze and whether or not we are at a turning point. This can be show in the form: (X,Y,T) where T represents turning point.

Initial State:

The initial state is now (max(X)/2, max(Y)/2, T). We are in the middle of the maze at either a turning point or non-turning point.

Actions:

There is one possible action for the robot, now. Moving. The robot can move(). move() can be executed and the robot will continue to move until a turning point is reached.

Transition Model:

Beginning in the initial state, there is one transition function for our one action.

(current\_state(X1,Y1,T), move()) = new\_state(X2,Y2,T)

Goal Test:

For any valid maze there should be at least 1 goal state. That is there should be some position in the set of possible positions that is defined the goal position. The goal state is this position with the robot in any orientation. The goal test will check to see that the robot is in a goal position, if it is then the task is complete.

Path Cost:

Every move will cost 1.

The state space must now be updated. We have the set P of possible positions for the robot. This being said, we do not need to keep track of the direction any more because the robot is turned for us at each turning point. Due to this we only care about the set of possible positions P.

The state space is now cardinality of P: |P|

**Part 4: List three ways that simplifications were made through abstractions:**

1. Direction was reduced to purely cardinal directions rather than angles

2. The robot will stop before hitting a wall

3. The robot will move a fixed distance x