Robert Watkins

CS 482

Problem Set [0]

1. Using the formulations outline in Toy problems in the book
   1. **States:**

The state is determined by the location of the robot and the current direction it faces. The robot can face north, south, east, or west.

Since we can assume each time the robot is moved it is moved the space of one block, we can determine that the maze is composed of n such blocks. Since the robot can turn in 4 directions, or 4 decisions can be made per block, **our state space is 4n**.

**Initial State:**

The initial state is given that the robot is in the center of the maze facing north.

**Actions:**

There are 5 total actions that can be performed. Turn north, turn south, turn east, turn west, and move in facing direction

**Transition model:**

The robot is able turn in any direction within any block but can only move in the facing direction until a wall is encountered.

**Goal test:**

This checks whether the robot is on a block outside of the maze area

**Path cost:**

We can consider that turning the robot has no real cost associated with it, therefore each block traversed can be considered a path cost of one and the total blocks taken to reach goal can be considered the total cost.

* 1. **States:**

The state is determined by the location of the robot, the current direction it faces, and whether or not the robot is at an intersection.

As before, the robot can face north, south, east, or west. The difference being that turning is not necessary unless the robot is at an intersection, otherwise the robot can continue moving in its current direction. We can again determine the maze is composed of n blocks and in addition the maze also has i intersections. So now we have two parts determining our state space. States in which we can turn in any 4 directions: 4i and states where we can only turn 2 directions along the current path of travel (not at an intersection): 2(n-i). **So the state space is 4i + 2(n-i).**

**Initial State:**

The initial state is given that the robot is in the center of the maze facing north.

**Actions:**

There are 5 total actions that can be performed. Turn north, turn south, turn east, turn west, and move in facing direction

**Transition model:**

Now the robot should only turn if it is in a block that has an intersection of 2 or more paths. If so the robot can turn in any direction to choose any of these paths, otherwise it may only continue to face its current direction or turn around.

**Goal test:**

This checks whether the robot is on a block outside of the maze area

**Path cost:**

We can consider that turning the robot has no real cost associated with it, therefore each block traversed can be considered a path cost of one and the total blocks taken to reach goal can be considered the total cost.

* 1. **States:**

The state is determined by the location of the robot and whether or not the robot is at an intersection.

This is similar to part b only that we don’t need to move the robot, only turn it at an intersection, i. If we only need to perform an action when a turning point is encountered that means we only need to know when we hit an intersection. In addition **we do not need to keep track of the robot’s orientation** as it doesn’t matter in what direction it is currently traveling. **So the state space just becomes i**.

**Initial State:**

The initial state is given that the robot is in the center of the maze facing north.

**Actions:**

There are 4 total actions that can be performed. Turn north, turn south, turn east, and turn west, movement of the robot itself has been automated.

**Transition model:**

Now the robot should only turn if it is in a block that has an intersection of 2 or more paths. If so the robot can turn in any direction to choose any of these paths, otherwise it may only continue to face its current direction or turn around.

**Goal test:**

This checks whether the robot is on a block outside of the maze area

**Path cost:**

We can consider that turning the robot has no real cost associated with it, therefore each block traversed can be considered a path cost of one and the total blocks taken to reach goal can be considered the total cost.

* 1. + The robot knows when a wall has been encountered.
     + The robot can only move north, south, east, or west and not diagonally in any way.
     + The maze is solvable, that is there are no instances where the robot will become stuck.

1. Assumption: Robot can only move around the outside of the polygons and not through them.
   1. If we are moving strictly from vertex to vertex then the shortest path between two vertices will always be a straight line. If we are to reach our goal we must travel between several polygons in effect linking them together with our path

**State Space:**

A state is the current vertex the robot is at. So the state space is the set of vertices, V.

**Action:**

Travel to a neighboring vertices that is closest to the goal.

* 1. An admissible heuristic would be the distance from a vertex to the goal in a straight line, disregarding traveling from vertex to vertex, this underestimates path cost as a connected path through vertices will surely be longer than a single path disregarding obstacles.
  2. Since we have to maintain a certain distance away from objects I assume this means from vertices as well, as such the only change would be we would have to be e centimeters from each vertices but the heuristic itself would not need to change to remain admissible, a straight line to the goal from the modified points still underestimates the actual path.
  3. A rough path that perhaps didn’t cut straight to the goal but favored shapes with orientations towards the goal relative to our current vertex.