CS 7750: Solutions to homework 2

Chanmann Lim

September 23, 2014

3.2

a.

States: all possible location and facing direction of the robot on the maze's map
Initial state: (centre, north)
Actions: { turn north, turn east, turn south, turn west, move forward }
Transition model: returns new (location, facing) of the robot
Goal test: checks whether the location of the robot is out of the maze
Path cost: each step costs 1, path cost = total number of step taken

State space: $4 \times$ *available squares on the maze*

b.

Actions: { turn north, turn east, turn south, turn west, move to intersection }
State space: $4\times$(*four intersection squares*)$+3\times$(*three intersection squares*)$+2\times$(*two intersection squares*)$+$ (*corridor without intersection squares*)

c.

Actions: { turn north, turn east, turn south, turn west, move until turning point }
State space: $4 \times$ *the number of turing points*

Now, it is not necessary to keep track of the robot's orientation since the action of turning robot's will not happen before reaching the turning point on the maze.

d. There are some details from the real world have been abstracted to simplify the problem formulation such:

1. The road condition of the maze (without crater, sand or water which will increase the complexity in moving)
2. How the robot move (by wheel, leg or fly) is abstracted
3. Turning precision. We assume the robot will always in the middle of the corridor when turning so that it will not hit the side when moving

3.3

a.

States: cities map with location of both friends
Initial state: (city_1, city_2)
Actions: move each friend to neighboring city
Transition model: returns current cities of friend 1 and friend 2 after moving
Goal test: checks whether if city_1 and city_2 is the same city
Path cost: step cost = max(d1(i, j), d2(i, j)), path cost = total step cost

b. $D(i,j)$ and $D(i,j)/2$ heuristic functions are admissible since both functions yield the result that is never over estimate the real distance, in other word the distance to the goal from n "$h^*(n)$" is always larger than the straight-line distance "$D(i,j)$" thus larger than "$D(i,j)/2$". However, there is no way to proof that $h^*(n)$ will be larger than heuristic function $h(n) = 2 \cdot D(i,j), thusitisnotanadmissibleheuristicfunction.$

c. Yes, if there are only 2 cities A and B (connected) and each friend is in one city, they will go in the opposite direction to see each other but will never meet.

d. Yes, suppose there are only 2 cities (connected) but one city has an extra circular route connected to itself, then one friend can go from $A->A$ and other can go from $B->A$ so that they can meet each other. In this scenario, one friend has to visit city A twice.

3.5

By counting the possibilities of placing one queen at a time on each column (or row) we can derive that there at least n, $(n-3)$, $(n-6)$, $(n-9)$ ... possibilities for column (or row) 1, 2, 3, 4 ... respectively when n is a large number. So the number of possibilities is

$$A = n \cdot (n-3) \cdot (n-6) \cdot (n-9) \ldots$$

Let

$$B = (n-1) \cdot (n-4) \cdot (n-7) \ldots$$
$$C = (n-2) \cdot (n-5) \cdot (n-8) \ldots$$

$$A \cdot B \cdot C = n \cdot (n-1) \cdot (n-2) \cdot (n-3) \cdot (n-4) \cdot (n-5) \cdot (n-6) \cdot (n-7) \cdot (n-8) \cdot (n-9) \ldots$$
$$A \cdot B \cdot C = n!$$

Since A, B, and C should have the same term to n! and $A > B$ and $A > C$

$$A^3 > n!$$
$$A > \sqrt[3]{n!}$$

So that the state space has at least $\sqrt[3]{n!}$ states.

3.6

a.

States: regions with colors but no repeating color next to each others

Initial state: regions without any color on the map

Actions: fill any color to a region on the map such that the adjacent regions does not have the same color

Transition model: returns the map with a color is filled in the specified region

Goal test: checks whether all regions are colored and no two adjacent regions have the same color

Path cost: each region coloring costs 1, path cost = total regions filled with color

b.

States: monkey position, crate 1 and crate 2 status(stacked?)
Initial state: the monkey is on the ground and 2 crates are not stacked
Actions: { stack, climb }
Transition model: returns the monkey position and 2 crates' status
Goal test: checks whether if the monkey reaches the banana $height_{(monkey)} \geq height_{(banana)}$
Path cost: each action costs 1, path cost = the number of actions performed

3.15

a.

b.

Breath first: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
Depth-limit(3): [1, 2, 4, 8, 9, 5, 10, 11]
Iterative deepening: [1, {1, 2, 3}, {1, 2, 3, 4, 5, 6, 7}, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}]

c. The bidirectional search will work well in this problem since the graph is a tree and there is only one predecessor of any states and the backward search can traverse directly to the root.

Branching factor for forward direction (initial ->goal): 2
Branching factor for backward direction (goal ->initial): 1

d. Yes, the backward direction(goal ->initial)'s branching factor is 1 and this indicates that there will be no search required in that direction since there is only one route mapped from state 1 to any goal state in the backward direction.

e.

Data: goal: the goal number(integer)
Result: solution: the solution to the goal number
$i \longleftarrow goal$;
$solution \longleftarrow [1]$;
while $(i/2) \neq 0$ **do**
  $solution \longleftarrow INSERT(i, 1)$;
  $i \longleftarrow i/2$;
end
return *solution*;