

Part 1: a. Since A^* follows a path that has the lowest expected total cost, it will move to East rather than North to minimize the total expected cost.

$$g(E) = 1, h(E) = 2 \rightarrow f(E) = 1 + 2 = 3$$

$$g(N) = 1, h(N) = 4 \rightarrow f(N) = 1 + 4 = 5$$

b. The agent will not get stuck in an infinite loop because we do not visit a grid that has already been expanded in that iteration. In this algorithm, we move that particular state to the closed list. Whenever the path traversed by the agent is blocked, we leave the path and go to the next unvisited state in the open list which has minimum expected cost (f-value). This makes sure we cover all the states in the grid. Unless there are infinite number of nodes, the agent (A^* algorithm) would find the goal state or would discover it is impossible to reach the goal in finite time (when the agent is separated from target by blocked cells).

Part 4: Consider the coordinates of goal state G be (G_x, G_y) . Let cell A be (x_1, y_1) . The estimated path cost from cell A to goal G (A-G) is

$$|G_x - x_1| + |G_y - y_1|.$$

Now assume the agent travels to the goal through a cell $B(x_2, y_2)$ which lies outside the path (A-G). The new path cost A-B-G is

$$|G_x - x_2| + |G_y - y_2| + |x_2 - x_1| + |y_2 - y_1|$$

Compare the two path costs, A-G and A-B-G.

$$|G_x - x_2| + |x_2 - x_1| \rightarrow |G_x - x_1|$$

(This is always true for any 3 values)

$$|G_y - y_2| + |y_2 - y_1| \rightarrow |G_y - y_1|$$

(This is also always true for any 3 values) Combining the two inequalities prove that estimated path cost

$$(A - B - G) \rightarrow (A - G).$$

So there can't be a shorter path to the goal through B . Therefore the initial path (A-G) estimated by the heuristic is the shortest path. Therefore Manhattan distances are consistent. From the proof, we have

$$(B - G) + (A - B) = (A - B - G) \rightarrow (A - G)$$

by triangle inequality.

Part 6:

We suggest the following methods for decreasing the memory usage of our implementation.

We can use a method when creating the states which does not create all the states initially. We only create a state if it's going to be explored by the algorithm.

Another possibility is prevent storing f-value and h- values, which can both be calculated easily. The g-values can also be calculated by following tree-parents until reaching current start state. However, since this will increase the computation costs considerably, we do not include this improvement in our calculation.

Another improvement could be instead of storing the counter as search variable, a boolean variable can be used to check if the state is visited in the current iteration.