CS440: Assignment 1 Write-Up

# Ramaseshan Parthasarathy, Vamshikrishnan Balakrishnan, Benjamin Yan

**Part 1:**

a) In figure 8, the agent would move east instead of north, because the cell east of the agent is closer to the goal than the cell north of the agent. If we are computing h(x) using Manhattan distances, the h(x) value for the cell east of the agent is 3. While the h(x) for the cell north of the agent is 4. If we are computing h(x) using straight-line distances the h(x) value for the cell east of the agent is 3. And the h(x) value of the cell north of the agent is sqrt(17), which is approximately 4.123. The g(x) values for the cell to the east of the agent and the cell to the north of the agent are 1 and 1 respectively. Therefore, if we are to compute h(x) using Manhattan distances the f(x) value of the cell to the east of the agent is 4 and the f(x) value of the cell to the north of the agent is 5. And if we are to compute h(x) using straight-line distances, the f(x) value of the cell to the east of the agent is 5, and the f(x) value of the cell to the north of the agent is 5.123. So no matter what you are using to compute h(x), the cell to the east of the agent is always preferable to the cell north of the agent, since the f(x) value of the cell east of the agent is always less than the f(x) value to the north of the agent.

b) If we visualize A\* search as a tree, and let all of the nodes of the tree be the cells of the environment, we would have a tree with a finite number of nodes. Cells are represented as nodes and their connections to their neighbors are represented by edges. The algorithm will be traversing the tree based on the smallest f(x) value among the nodes in the open list. In the worst case, we would have to traverse the whole tree before reaching the goal node. Since there are a finite number of nodes, traversing all of them must take a finite amount of time.