CS 480 - Homework #1 Solutions

Aman Gill Kyle Janssen

Spring 2014

1) Show the assertions

Assertion 1. *If $h_1(n), h_2(n)$ are admissible then $h(n) = max(h_1(n), h_2(n))$ is admissible.*

Proof. Let h_1, h_2 describe admissible heuristics. By definition, $h_1(n) \leq C(n)$ and $h_2(n) \leq C(n)$ where $C(n)$ is the actual cost to get from node n to the goal node. Without loss of generality, assume $h_1 \leq h_2(n)$ for some node $n*$.

if

$$h(n) = max(h_1(n), h_2(n))$$

then

$$h(n*) = h_2(n*)$$

since

$$h_2(n*) \leq C(n*)$$

$$h(n*) \leq C(n*)$$

and $h(n)$ is therefore admissible. □

Assertion 2. *If $h_1(n), h_2(n)$ are consistent then $h(n) = max(h_1(n), h_2(n))$ is consistent.*

Proof. Let h_1, h_2 describe consistent heuristics. By definition,

$$h_1(n_j) \leq C(n_{j+1}, n_j) + h_1(n_{j+1})$$
$$h_2(n_j) \leq C(n_{j+1}, n_j) + h_2(n_{j+1})$$

where $C(n_{j+1}, n_j)$ is the cost to get from node n_j to n_{j+1}. We will also define

$$h(n) = max(h_1(n_j), h_2(n_j))$$

There will be two cases: one where one heuristic is greater than both nodes, and one where the greater heuristic switches between nodes.

Case 1: Without loss of generality, assume

$$h_1(n_j) \leq h_2(n_j)$$
$$h_1(n_{j+1}) \leq h_2(n_{j+1})$$

then

$$h(n_j) = h_2(n_j) \text{ and } h(n_{j+1}) = h_2(n_{j+1})$$

Since h_2 is known to be consistent, h is also consistent.

Case 2: Without loss of generality, assume

$$h_1(n_j) \leq h_2(n_j)$$
$$h_2(n_{j+1}) \leq h_1(n_{j+1})$$

then

$$h(n_j) = h_2(n_j) \text{ and } h(n_{j+1}) = h_1(n_{j+1})$$

additionally,

$$C(n_{j+1}, n_j) + h_2(n_{j+1}) \leq C(n_{j+1}, n_j) + h_1(n_{j+1})$$

since h_2 is consistent,

$$h_2(n_j) \leq C(n_{j+1}, n_j) + h_2(n_{j+1}) \leq C(n_{j+1}, n_j) + h_1(n_{j+1})$$

Therefore, by substitution, h is consistent.

□

2) Exhibit the search tree

3) Justify a true/false answer

a) False. For a weighted, directed graph, increasing all of the weights of the edges by some amount A would cause the cost of the shortest path from node s to node g to increase by an amount proportional to the number of steps in that path. If the shortest path is not also the path with the fewest steps and A is of sufficient magnitude, the path with the fewest steps from s to g would become the shortest path.

b) True. When best first search is implemented using an admissible heuristic function, it is the A* algorithm, but unless the heuristic function is also consistent, the possibility remains that a node may be discovered more than once and its cost improved.

4) Pancake sorting problem

a) The heuristic function $h(x) = 1/2$ * the number of breakpoints is admissible. Each step between states costs 1, and each step away from the goal node can increase the number of breakpoints by 2, or $h(x)$ by 1. Because the heuristic and the cost to get to the goal increase at the same rate as we move further from the goal in our state space, $h(x)$ can never overestimate cost, and is therefore admissible.