3010 Artificial Intelligence: Assignment 2
Due: 5 pm, Monday, June 3, 2019

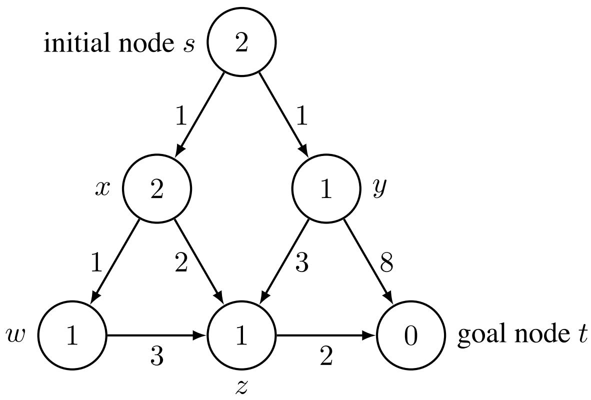
Write a report answering Questions 1–2. (Note: questions continue to the back of the page). Post the report in the drop-in box in front of Information Science Administration Office, by no later than 5 pm, June 3.

Note: We use the following terminology:

- A heuristic function h is said to be ***admissible*** in a state space graph if $0 \le h(v) \le h^*(v)$ holds for every node v in the graph, where $h^*(v)$ is the cost of the cheapest path from node v to a nearest goal node.
- A heuristic function h is said to be ***monotone*** in a state space graph if (i) for every edge (v, u) in the graph, $h(v) \le h(u) + c(v, u)$ holds, and (ii) $h(t) = 0$ holds for every goal node t.

Question 1

Consider the state space graph shown below. This graph has six nodes (s, x, y, z, w, t), with an initial node s and a goal node t. The number inside each node represents the value of the heuristic function h at the node, and the number next to each edge represents its cost. For instance, $h(s) = 2$, $h(w) = 1$, and the cost of moving from y to t is $c(y, t) = 8$.



Now answer the following questions.

1. Is this heuristic evaluation function h monotone? Explain your answer.
2. Is h admissible? Explain your answer.
3. Suppose we run the A* algorithm of Figure 1 on this graph. In each iteration of lines 7–14 of function AStar (on the left-hand side of the figure),
   - show which nodes are in OPEN and CLOSED when line 8 is executed, as well as their g- and f-values; and
   - show which node is chosen as v on line 10.

```
function AStar(s)
    OPEN ← new PriorityQueue_f
    g[s] ← 0
    f[s] ← h(s)
    Insert_f(OPEN, s)
    CLOSED ← ∅
    loop do
        if IsEmpty(OPEN) then
            return "failure"
        v ← DeleteMin_f(OPEN)
        CLOSED ← CLOSED ∪ {v}
        if IsGoal(v) then
            return Solution(v, s)
        Expand(v)
```

```
procedure Expand(v)
    foreach u ∈ Succ(v) do
        if u ∉ OPEN ∪ CLOSED then
            g[u] ← g[v] + c(v, u);  f[u] ← g[u] + h(u)
            Parent[u] ← v
            Insert_f(OPEN, u)
        else if u ∈ OPEN then
            if g[v] + c(v, u) < g[u] then
                g[u] ← g[v] + c(v, u);  f[u] ← g[u] + h(u)
                Parent[u] ← v
        else
            if g[v] + c(v, u) < g[u] then
                g[u] ← g[v] + c(v, u);  f[u] ← g[u] + h(u)
                Parent[u] ← v
                CLOSED ← CLOSED\{u}
                Insert_f(OPEN, u)
```

Figure 1: A* algorithm. OPEN, CLOSED, Parent, g, and f are global variables. See the lecture slides for more detail.

Answer

1. h is monotone, because $h(t) = 0$, and for each edge (v, u), $h(v) \leq c(v, u) + h(u)$ indeed holds, as shown in the following table.

Edge	v	u	$h(v)$	$h(u)$	$c(v,u)$	$h(u) + c(v,u)$	$h(v) \leq h(u) + c(v,u)$
(s,x)	s	x	2	2	1	3	True
(s,y)	s	y	2	1	1	2	True
(x,z)	x	z	2	1	2	3	True
(y,z)	y	z	1	1	3	4	True
(y,t)	y	t	1	0	8	8	True
(z,t)	z	t	1	0	2	2	True
(w,z)	w	z	1	1	3	4	True

2. Since h is monotone, it is also admissible.

3. See the following table.

	OPEN	CLOSED	$g[s]/f[s]$	$g[x]/f[x]$	$g[y]/f[y]$	$g[z]/f[z]$	$g[w]/f[w]$	$g[t]/f[t]$	chosen as v
1	s		0 / 2	/	/	/	/	/	s
2	x, y	s	~~0~~ / ~~2~~	1 / 3	1 / 2	/	/	/	y
3	x, z, t	s, y	~~0~~ / ~~2~~	1 / 3	~~1~~ / ~~2~~	4 / 5	/	9 / 9	x
4	z, w, t	s, x, y	~~0~~ / ~~2~~	~~1~~ / ~~3~~	~~1~~ / ~~2~~	3 / 4	2 / 3	9 / 9	w
5	z, t	s, x, y, w	~~0~~ / ~~2~~	~~1~~ / ~~3~~	~~1~~ / ~~2~~	3 / 4	~~2~~ / ~~3~~	5 / 5	z
6	t	s, x, y, w, z	~~0~~ / ~~2~~	~~1~~ / ~~3~~	~~1~~ / ~~2~~	~~3~~ / ~~4~~	~~2~~ / ~~3~~	5 / 5	t

Question 2

Explain whether each of the following statements is true or false.

1. "If a heuristic function is not admissible, then it is not monotone."

2. "Let h be a monotone heuristic function, and and let $k > 1$. Now define $h'(v) = kh(v)$ for every node v. If h' is admissible, then h' is monotone as well."

3. "Let $h_1(v)$ and $h_2(v)$ be two monotone heuristic functions for a graph, and let $h''(v) = \max(h_1(v), h_2(v))$ for every node v. Then, h'' is also monotone."

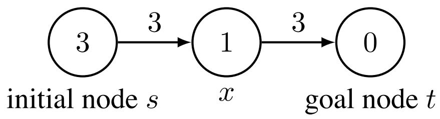
(Note: $\max(a, b)$ is a function that returns the larger of the two values a and b.)

4. "Let $h_1(v)$ and $h_2(v)$ be two monotone heuristic functions for a graph, and define $h'''(v) = h_1(v) + h_2(v)$ for every node v. If h''' is admissible, then h''' is monotone.

Answer

1. True. The statement is the contraposition of the preoperty described in the lecture, "All monotone heurisitc functions are admissible."

2. False. As a counterexample, consider the following graph.



Let the heurstic estimates be $h(s) = 3$, $h(x) = 1$, and $h(t) = 0$. It is easy to see that h is monotone. Now consider $h'(v) = 2h(v)$, i.e., $k = 2$. Then, $h'(s) = 6$, $h'(x) = 2$, $h'(t) = 0$. h' is admissible, as the actual shortest path costs are $h^*(s) = 6, h^*(x) = 3$, and $h^*(t) = 0$, and hence $h'(v) \le h^*(v)$ holds for every node v. However, h' is not monotone, because $h'(s) = 6 > 2 + 3 = h'(x) + c(s, x)$.

3. True. Since h_1 and h_2 are both monotone,

$$h_1(v) \le h_1(u) + c(v, u),$$
$$h_2(v) \le h_2(u) + c(v, u),$$

for every edge (v, u). Because $a \le \max(a, b)$ and $b \le \max(a, b)$,

$$h_1(v) \le \max(h_1(u), h_2(u)) + c(v, u),$$
$$h_2(v) \le \max(h_1(u), h_2(u)) + c(v, u).$$

It follows that

$$\max(h_1(v), h_2(v)) \le \max(h_1(u), h_2(u)) + c(v, u),$$

and thus

$$h''(v) \le h''(u) + c(v, u), \quad (1)$$

for every edge (v, u). Also, the monotonicity of h_1 and h_2 implies $h_1(t) = h_2(t) = 0$ for every goal node t, and hence $h''(t) = \max(h_1(t), h_2(t)) = 0$. This, together with Eq. (1), shows that h'' is also monotone.

4. False. The counterexample for Statement 2 above also provides the counterexample for this statement. To see why, let h_1 and h_2 both be the monotone heuristic function h in the counterexample. Then, we have $h'''(v) = h_1(v) + h_2(v) = h(v) + h(v) = 2h(v) = h'(v)$ for each node v. As shown above, h' is not monotone, and hence h''' is not monotone, either.