6th Edition, p. 147

Prop. If the h component of an evaluation function f is monotone, then f is monotonically nondecreasing.

Proof. Let $n_1, \ldots, n_k$ be some search path in the state space. By definition of f, for any i,

$$f(n_i) = g(n_i) + h(n_i) \qquad (1)$$

$$f(n_{i+1}) = g(n_{i+1}) + h(n_{i+1}) \qquad (2)$$

From (1) we have

$$f(n_i) = g(n_i) + h(n_i) - h(n_{i+1}) + h(n_{i+1}) \qquad (3)$$

Note that $g(n_{i+1}) = g(n_i) + cost(n_i, n_{i+1})$. Thus, from (2) we have

$$f(n_{i+1}) = g(n_i) + cost(n_i, n_{i+1}) + h(n_{i+1}) \qquad (4)$$

Since h is monotone, we have

$$h(n_i) - h(n_{i+1}) \leq cost(n_i, n_{i+1}).$$

This gives

$$g(n_i) + (h(n_i) - h(n_{i+1})) + h(n_{i+1}) \leq g(n_i) + cost(n_i, n_{i+1}) + h(n_{i+1}).$$

From (3) and (4), this gives

$$f(n_i) \leq f(n_{i+1}).$$

Thus, as i increases, so does $f(n_i)$.

6th Edition, p. 148

Prop. If the h component of an evaluation function f is monotone, then the associated Algorithm A is admissible.

Proof. Let $s_1, \ldots, s_g$ be a sequence of states in a search path, where s_1 is the start state and s_g is the goal. By the monotone property, we have

$$h(s_1) - h(s_2) \leq cost(s_1, s_2)$$
$$h(s_2) - h(s_3) \leq cost(s_2, s_3)$$
$$\vdots$$
$$h(s_{g-1}) - h(g) \leq cost(s_{g-1}, s_g)$$

Summing both sides gives

$$h(s_1) - h(s_g) \leq cost(s_1, s_g) \qquad (1)$$

Note that $cost(s_1, s_g) = h^*(s_1)$, by definition of h^*. Also, $h(s_g) = 0$, since h is monotone. Thus from (1) we have

$$h(s_1) \leq h^*(s_1).$$

This means that our algorithm A is actually algorithm A^*, in which case A is admissible by the result cited on p. 146.

6th ed, p. 158

$$T = B + B^2 + \cdots + B^L$$
$$= B(1 + B + \cdots + B^{L-1})$$
$$= B(1 + B + \cdots + B^{L-1} + B^L - B^L)$$
$$= B(1 + T - B^L)$$
$$= B + BT - B^{L+1}$$
$$T - BT = B - B^{L+1}$$
$$T(1-B) = B - B^{L+1}$$
$$T = \frac{B - B^{L+1}}{1-B}$$
$$= \frac{B - B^{L+1}}{1-B} \cdot \frac{-1}{-1}$$
$$= \frac{B^{L+1} - B}{B-1}$$
$$= \frac{B(B^L - 1)}{B-1}$$