Admissibility and Optimality of A^* Search

Note: Graph nodes with different subscripts are for different nodes in the search graph (e.g., n_1, n_2 are two different nodes in the set N of G). Meanwhile, same graph nodes with different superscripts are for their different appearances in the locations of the search tree (e.g., n_2' and n_2'').

Definition of Admissibility of $\mathbf{A}^*$ The A^* search is admissible iff, as long as solutions exist, an optimal solution (i.e., a shortest path from a start node s to a goal node g) will be found by A^*, even if the search tree is infinite (i.e., graphs with cycles).
Theorem: The search strategy A^* is admissible if

- the branching factor of the search tree is finite.
- edge costs are bounded above zero.
- the heuristic function $h(n)$ is a lower bound on the actual minimum cost of the shortest path from n to the goal node g.

Proof.

1. We will first prove that A^* always find a solution upon the settings above. If the search tree is with finite depth (no cycle), then the frontier F will not be trapped into infinite cycles, and all nodes/paths will be explored sooner or later, including the solution paths, since there exists at least one. Meanwhile, if the tree is infinite, it means some nodes will be inserted into and selected from F repeatedly. But each time the node, say n, is selected its *f-value* will be increased, since the actual cost of reaching n from s, which is $g(n)$, always increases. But this means a frontier node for the solution path will eventually be selected.

2. We prove that the 1st solution is optimal. Suppose we have an optimal path
$$(s, n_1, n_2, \ldots, n_k, g),$$
and $(s \to g')$ refers to a non-optimal solution. We know that $f(g') = g(g') + h(g') = g(g')$, which equals to the actual cost from s to g'. Since n_1 is a neighbor of s, it must be inserted into F. But $f(n_1) = g(n_1) + h(n_1)$ is less than or equals to $f(g)$, the minimal cost of the path to g from s. Hence, we have $f(n_1) < f(g')$, which means n_1 will be expanded before g'. In addition, n_2 as a neighbor of n_1, will be inserted into $F, \ldots$ Eventually, n_k will be inserted into F before g' is expanded. Since $f(n_k) < f(g')$, n_k will be expanded, leading to the insertion of g. Since $f(g) < f(g')$, finally the optimal path/nodes, g, will be selected before g'.