Assumptions:

$h(s)$ is an admissible heuristic $\Rightarrow$ $0 \leq h(s) \leq \text{length}(\text{path}(s,s_g))$

All actions have a positive cost

Let $\text{path}(s_i,s_g)$ be the path returned by A*.

Assume for the sake of contradiction that there exists a second path, $\text{path}(s_i,s_g')'$ such that

$\text{cost}(\text{path}(s_i,s_g')') < \text{cost}(\text{path}(s_i,s_g))$

Note that it we do not exclude the case where $s_g=s_g'$, in which case $\text{path}(s_i,s_g')'$ would be a shorter path to the same goal.

We first observe that the path length from a node to itself is 0.

$\text{length}(\text{path}(s,s)) = 0$

and thus the length of the shortest path from a goal state to a goal state must be 0.

$\text{path}(s_g,s_g)=0$

And because h is an admissible heuristic

$h(s_g)=0$

Therefore

$g(s_g) = \text{cost}(\text{path}(s_i,s_g)) + h(s_g) = \text{cost}(\text{path}(s_i,s_g))$

Because $\text{cost}(\text{path}(s_i,s_g')') < \text{cost}(\text{path}(s_i,s_g))$ we know that $g(s_g') < g(s_g)$

Let s_j be the first node on $\text{path}(s_i,s_g')'$ that was not expanded by A*. Because s_j's predecessor on path $\text{path}(s_i,s_g')'$ has been expanded we know that s_j must be on the priority queue when the A* algorithm returns. We next observe that

$g(s_j) = \text{path}(s_i,s_j) + h(s_j)$

Because h is admissible we know that $h(s_j) \leq \text{path}(s_i,s_j)$ which means that

$g(s_j) = \text{path}(s_i,s_j) + h(s_j) \leq \text{path}(s_i,s_j) + \text{path}(s_i,s_j)$

and $\text{path}(s_i,s_j) + \text{path}(s_i,s_j) = \text{path}(s_i,s_g')'$, so

$g(s_j) \leq \text{path}(s_i,s_g')'$

We also know that

$g(s_g) = \text{cost}(\text{path}(s_i,s_g))$ and

$\text{cost}(\text{path}(s_i,s_g')') < \text{cost}(\text{path}(s_i,s_g))$, so

$g(s_j) \leq g(s_g)$

Because A* selects nodes from a priority queue ordered by $g(s)$, we know that s_j cannot be on the priority queue when A* returns s_g because A* would have selected s_j instead of s_g. By contradiction $\text{path}(s_i,s_g')'$ cannot exist. Thus the path returned by A* must be the shortest path