Uniform Search Optimality

Proof

Let \( G \) be the goal nodes. 

\[ c(p_i) = \text{cost of path } p_i. \]

Let \( C(p_i) = \min \left\{ c(p_i), C(p_{i-1}) \cdot E(\text{cost}) \right\} \)

Each edge cost \( \geq 0 \).

Let \( p_{\text{cost}} = \text{path from } n_k \text{ to } G_i \).

If \( C(p_i) \) is not optimal, let

\[ p_{\text{optimal}} = \text{optimal path}. \]
Let $C_{opt} = \text{const} \left( p_k + p_{vk} \right)$

So $C(p_k) > C_{opt}$

$\geq C(p_k) + C(p_{vk})$

$\Downarrow$

$> 0$

So $C(p_k) > C(p_{vk})$

So algorithm should have picked $N_k$ instead of $G_k$—

a contradiction.
Tree Search (A* Search)

Let DS denote data structure.

Put start state in DS

Repeat until Solution is found:

1. Pick a node from DS such that it has lowest $f$ value among all nodes in DS

2. If node picked is goal then Solution found. Exit.

3. Otherwise generate its successors and push them in DS with their $f$ values.
Graph Search (A*)

CL - Closed list (or Expanded list)
DS - Data Structure.

Put start state in DS.

Repeat until solution is found.

1. Pick a node say n from DS such that f(n) is the lowest among all nodes in DS.

2. If n is goal then solution found & exit.

3. Otherwise:
   (i) Put n in CL.
   (ii) Generate successors of n.
   (iii) Only insert those successors of n that are not in DS.
Let $C_{opt}$ be the cost of optimal solution. DS is data structure. Also called open list or fringe list.

Lemma - If a node $n$ is chosen for expansion from DS then $f(n) \leq C_{opt}$.

Proof (Sketch) - $P_{opt}$ is path whose cost is $C_{opt}$.

\[ f(n) \leq f(n') = g(n') + h(n') \]

By admissibility $C(n', n) > h(n')$.

Therefore $f(n') \leq C_{opt}$ and $\therefore f(n) \leq C_{opt}$. 
Definition: heuristic \( h_2 \) is more informed than heuristic \( h_1 \), if \( \forall \) nodes \( v \), \( h_2(v) \geq h_1(v) \).

Theorem: A* Node Expansion

Let \( A_1 \) be the search tree constructed by A* using \( h_1 \). \(-\text{A*}(h_1)\)
\( A_2 \) be the search tree constructed by A* using \( h_2 \). \(-\text{A*}(h_2)\)

Theorem: Any node expanded by \(-\text{A*}(h_2)\) will be expanded by \(-\text{A*}(h_1)\).
Proof (Sketch)

Assume by induction that $A^*_s(h_i)$ expands all nodes expanded by $A^*(h_i)$ whose depth in $A_2 \leq k$.

Base If depth = 0 then node expanded in start start state in both $A_1$ & $A_2$.

Induction Step Assume inductive

Assume $n$ is expanded in $A_2$ but not in $A_1$.

Claim: $B_3$ in induction hypothesis $n$ exists in $A_1$ also.

\[ \text{depth} = k \]
$h$ is expanded in $A_2$ but not in $A_1$. \[ n \in D \subseteq A_1. \]

So when solution is found:

\[
 f(n) \leq \text{Cost} \in A_2
\]

i.e., \( h_2(n) + g(n) \leq \text{Cost} \)

But \( \text{Cost} \leq f(n) \in A_1 \) (since \( n \in D \subseteq A_1 \))

\[
 \leq h_1(n) + g(n)
\]

\[
 \therefore h_2(n) \leq h_1(n)
\]

contradiction.
Let's revisit uniform cost search.

Path cost based on $g$ monotonically.

\[ g(n') \geq g(n) \text{ if } n \text{ appears before } n' \text{ in path from } S. \]

This is because cost on edges are non-negative.
How do we ensure this if we use \( f(n) \)?

\[
\begin{align*}
\text{Start} & \quad \rightarrow \quad f(n) \\
\rightarrow & \quad n \\
\rightarrow & \quad c(n, n') \\
\rightarrow & \quad f(n') \\
\rightarrow & \quad \text{end State}
\end{align*}
\]

\[
f(n) = g(n) + h(n)
\]

\[
f(n') = g(n') + h(n')
\]

We want: \[ g(n) + c(n, n') + h(n') \]

\[
f(n) \leq f(n') \implies h(n) \leq c(n, n') + h(n').
\]