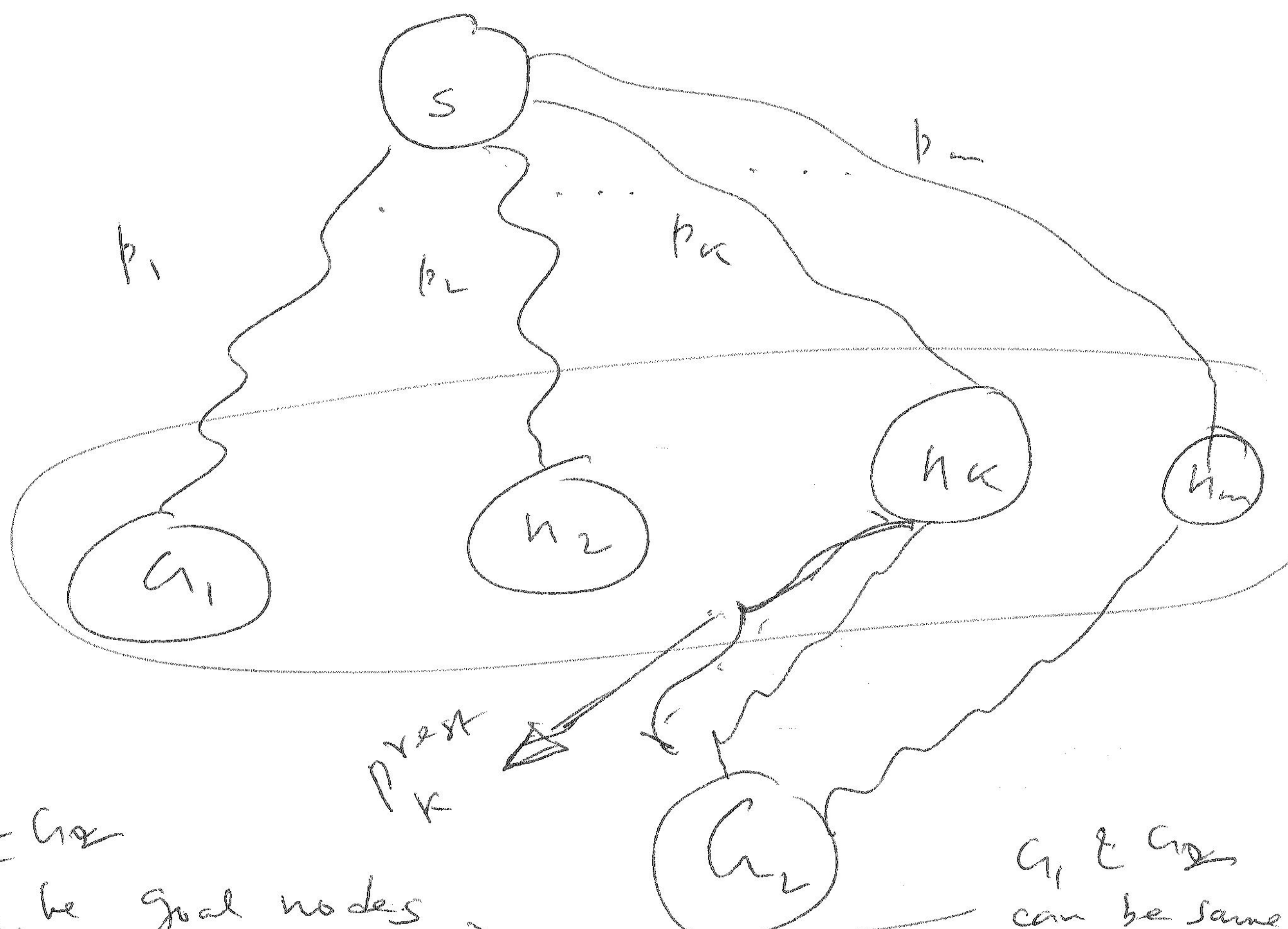
Uniform Search Optimality Proof



Let G_1 & G_2 be goal nodes. G_1 & G_2 can be same.

$C(p_i)$ = cost of Path p_i.

Let $C(p_1) = \min \{ C(p_1), C(p_2) \cdots C(p_m) \}$

each edge cost > 0.

Let p_k^{rest} = path from n_k to G_2

If $C(p_1)$ is not optimal let

$p_k + p_k^{rest}$ = optimal path.

Let $C_{opt} = \mathrm{Cost}(p_k + p_k^{rest})$

So $C(p_1) > C_{opt}$

$$> C(p_k) + C(p_k^{rest})$$

$$\downarrow$$

$$> 0$$

So $C(p_1) > C(p_k)$

So algorithm should have picked ~~p_k~~ p_k instead of G_1 — 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 ~~fn~~ f value among all nodes in DS

2. If node picked is goal then solution found. & exit.

3. Otherwise generate its successors & puth 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.

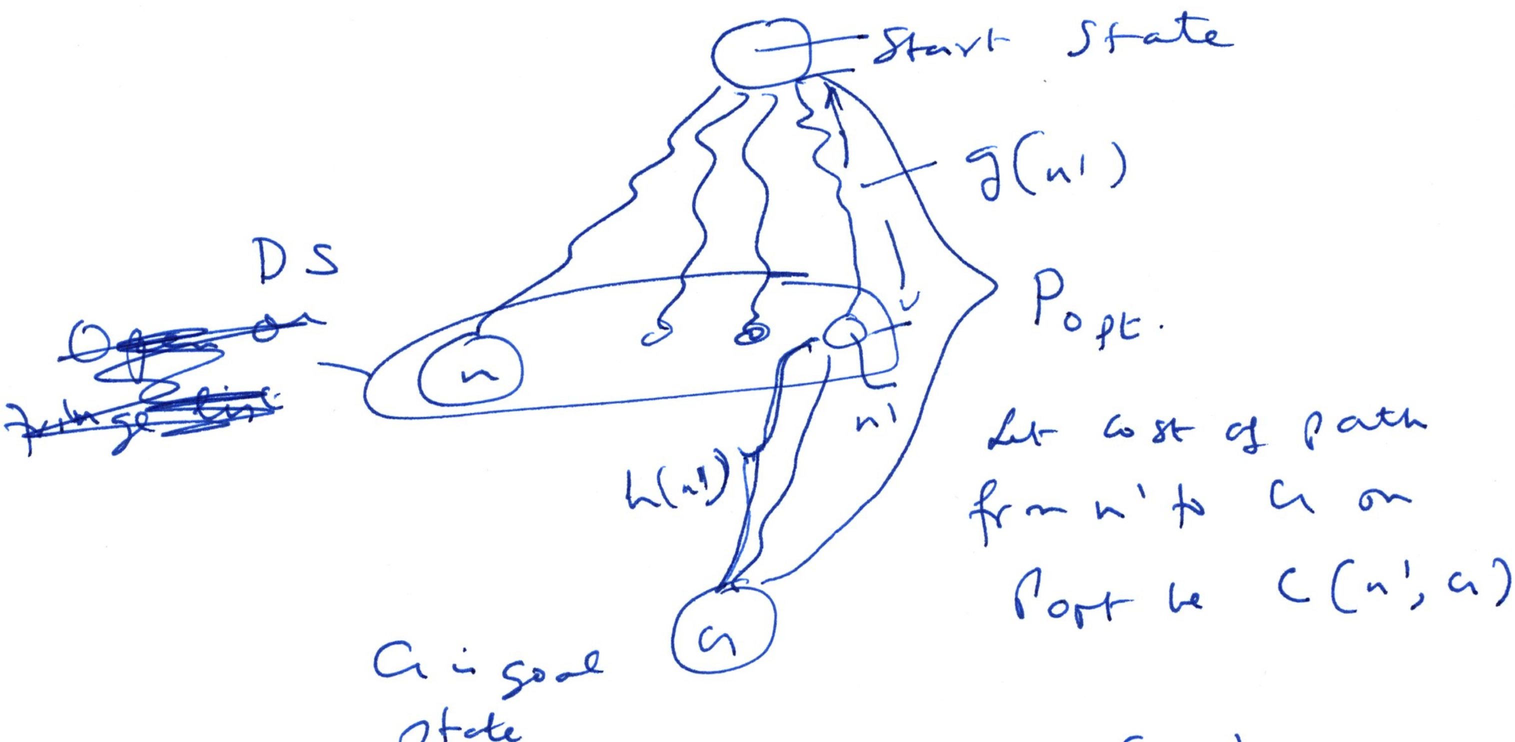
1. Pick a node say n from DS such that f(n) is the lowest among all nodes in DS.
2. If ~~node~~ n is goal then solution found & exit.
3. Otherwise:
   (i) Put n in CL.
   (ii) generate successors of n.
   (iii) Only insert those successors of n ~~that are not~~ in DS that are not in CL.

~~Open/Fringe~~

Let C_{opt} be cost of optimal ~~soluti~~ solution. DS is data structure. Also called open list or fringe list.

<u>Lemma</u> - If a node n is chosen for expansion from ~~DS~~ DS then $f(n) \le C_{opt}$.

<u>Proof (Sketch)</u>. P_{opt} is path whose cost is C_{opt}.



$$f(n) \le f(n') = g(n') + h(n')$$

By admissibility $C(n', G) \geq h(n')$

$\therefore f(n') \le C_{opt}$ & $\therefore f(n) \le C_{opt}$.

<u>Defn</u> heuristic h_2 is more ~~informed~~ informed than heuristic h_1 iff $\forall$ nodes n $h_2(n) > h_1(n)$

~~Theorem Any node~~

Let A_1 be the search tree constructed by A^* using h_1. – $A^*(h_1)$ & A_2 be the search tree constructed by A^* using h_2. – ~~A^*~~ $A^*(h_2)$

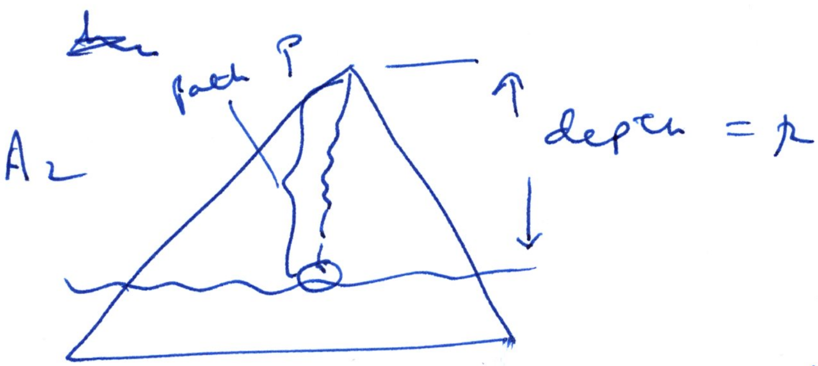
<u>Theorem</u> Any node expanded by $A^*(h_2)$ will be expanded by $A^*(h_1)$

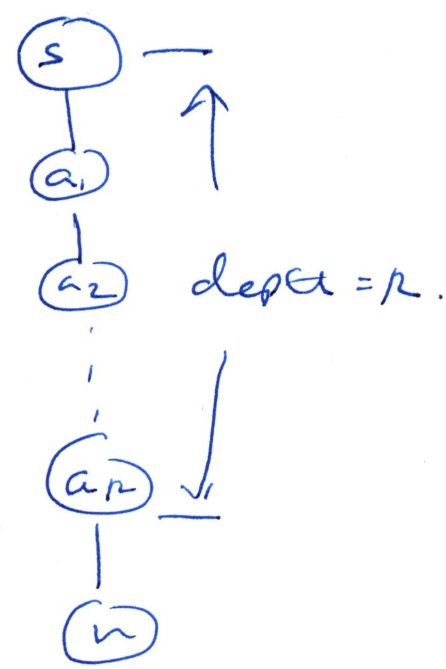
Proof (Sketch)

Assume by induction that $A^*_1(h_1)$ expands all nodes expanded by $A^*(h_2)$ whose depth in $A_2 \le k$.

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

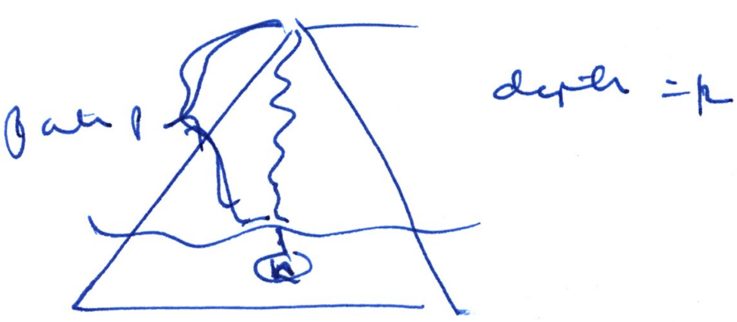
Induction step ~~Assume inductive~~





Assume n is expanded in A_2 but not in A_1.

Claim: By induction hypothesis p exists in A_1 also.



n is expanded in A_2 but not in A_1. ~~A_1 ii~~ n is in DS in A_1.

So when solution is found:

$$f(n) \le C_{opt} \text{ in } A_2$$

i.e., $h_2(n) + g(n) \le C_{opt}$

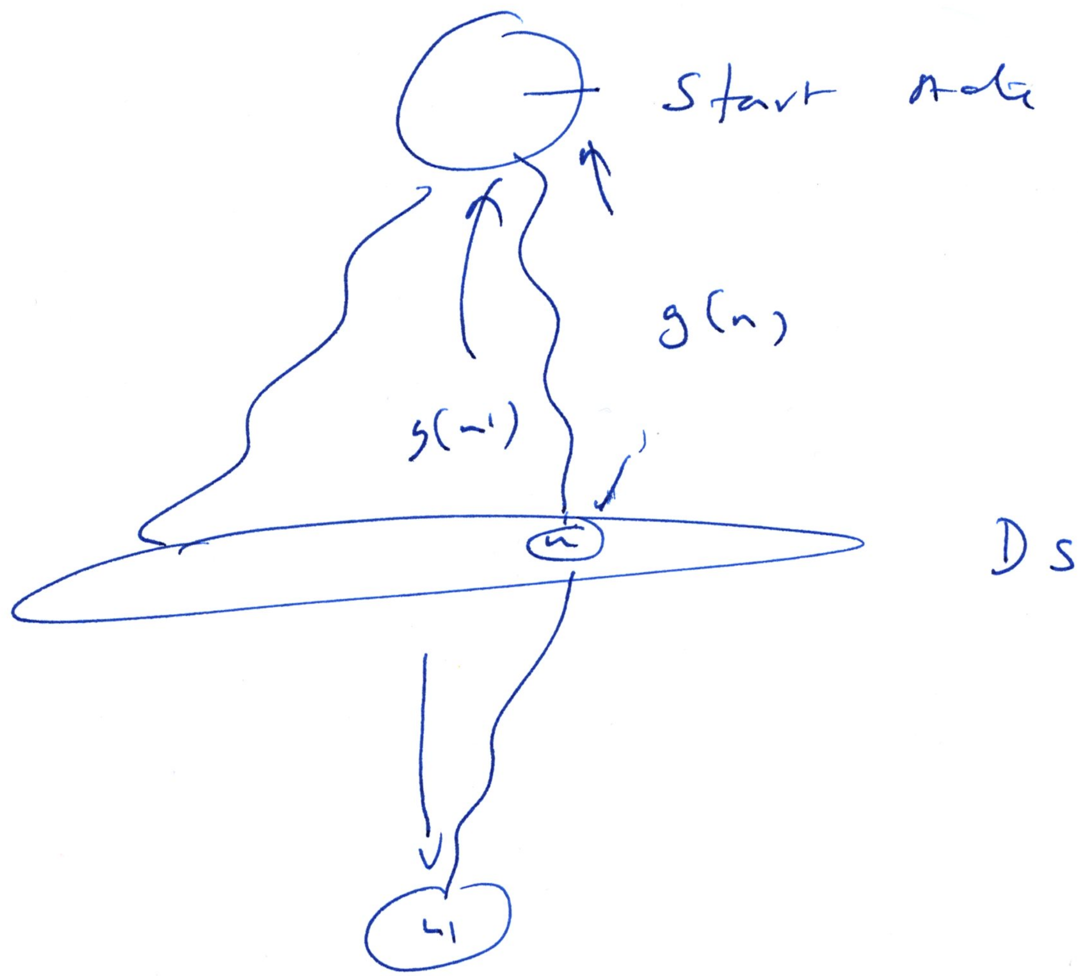
But $C_{opt} \le f(n)$ in A_1 (since n is in DS in A_1)

$$\le h_1(n) + g(n)$$

$$\therefore \; h_2(n) \le h_1(n)$$

contradiction.

Lets revisit Uniform cost search



path cost based on g monotonically increases.

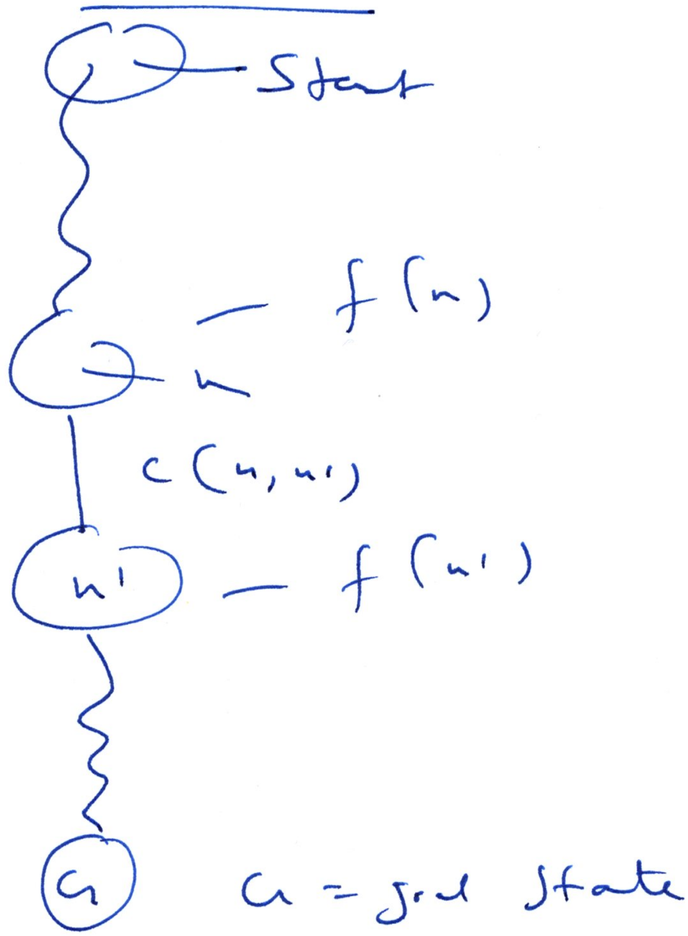
i.e. $g(n') \geq g(n)$ if

n appears before n' in path

from S. This is because

cost on edges are non-negative.

How do we ensure this if we use $f(n)$.



$$f(n) = g(n) + h(n)$$

$$f(n') = g(n') + h(n')$$

$$= g(n) + c(n, n') + h(n')$$

we want:

$$f(n) \le f(n')$$

$$\therefore \; h(n) \le c(n, n') + h(n').$$