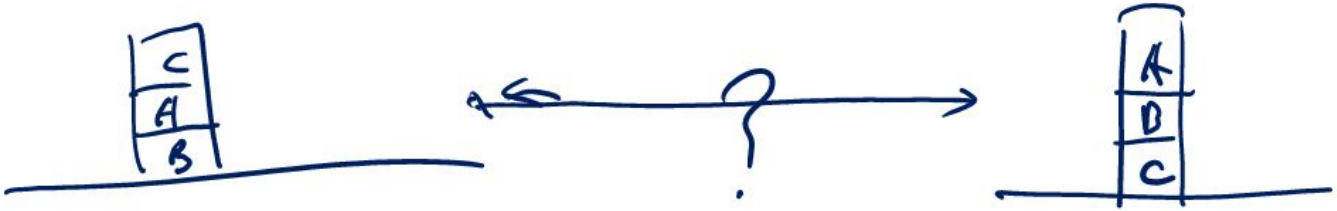
COMP-4475

All slides

W1ch2 lecture notes. Jan 11th

problem solving through searching on state space, block worlds, 8-puzzles, and Hanoi.

a. classical. AI. problem. : BLOCK WORLD.

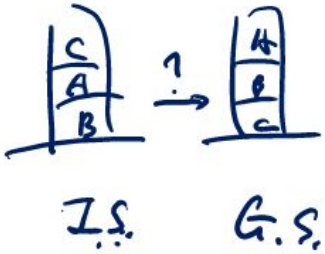


initial state.

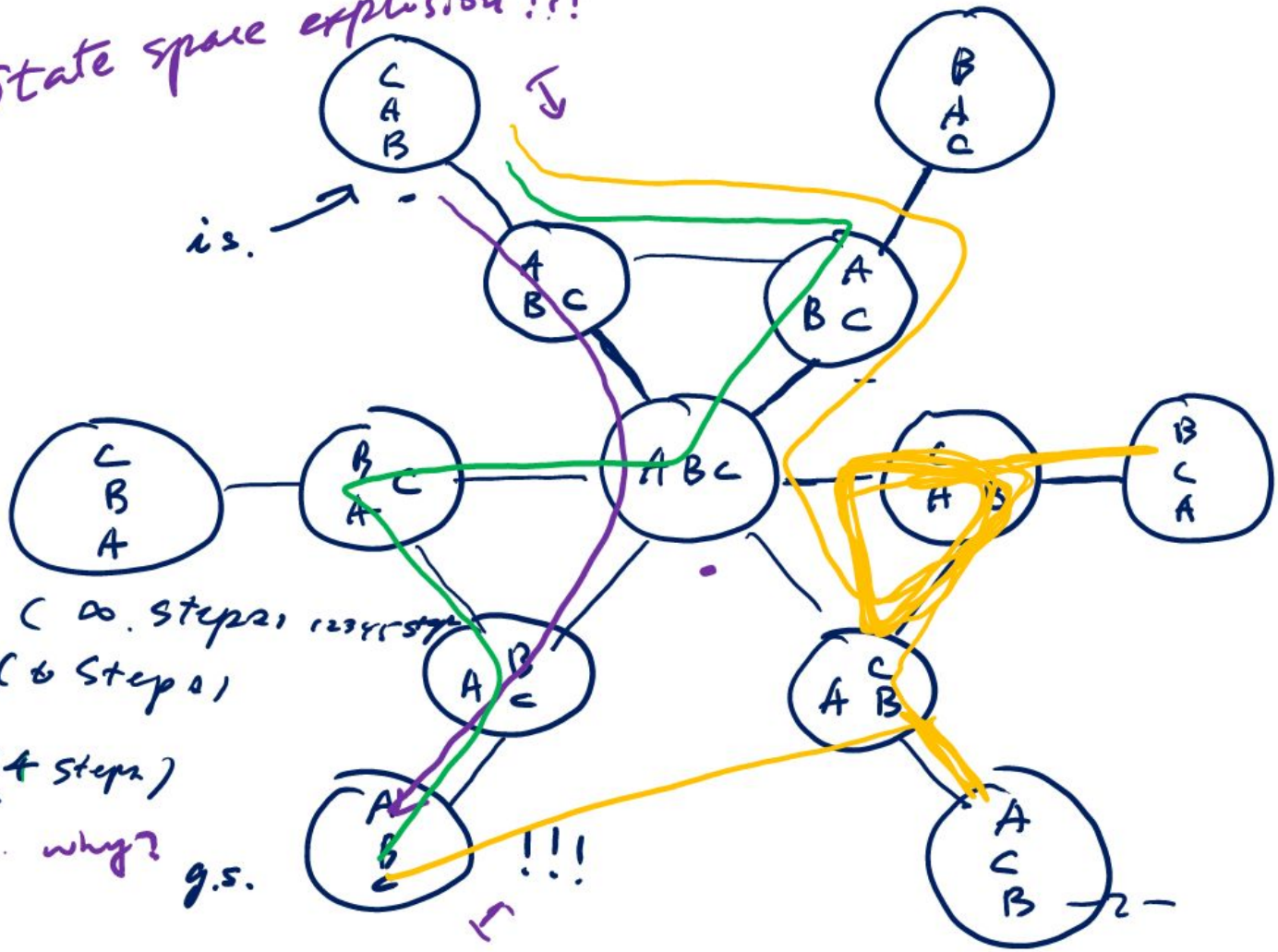
a sequence of actions

goal state.

Searching in <u>State-space</u>



State space explosion !!!



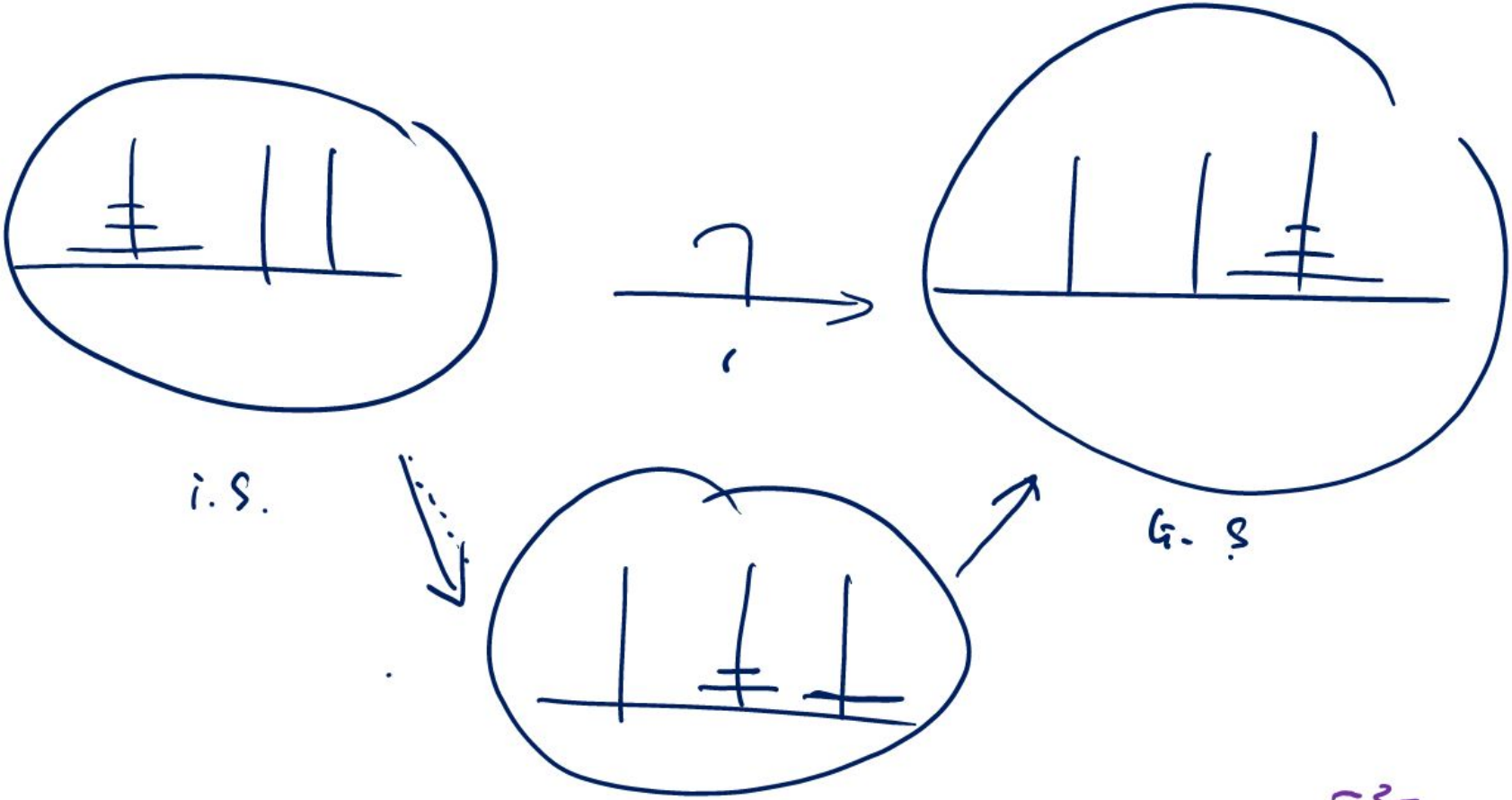
— ouch! (∞. steps, 1234567 step)

— fine! (6 Step s)

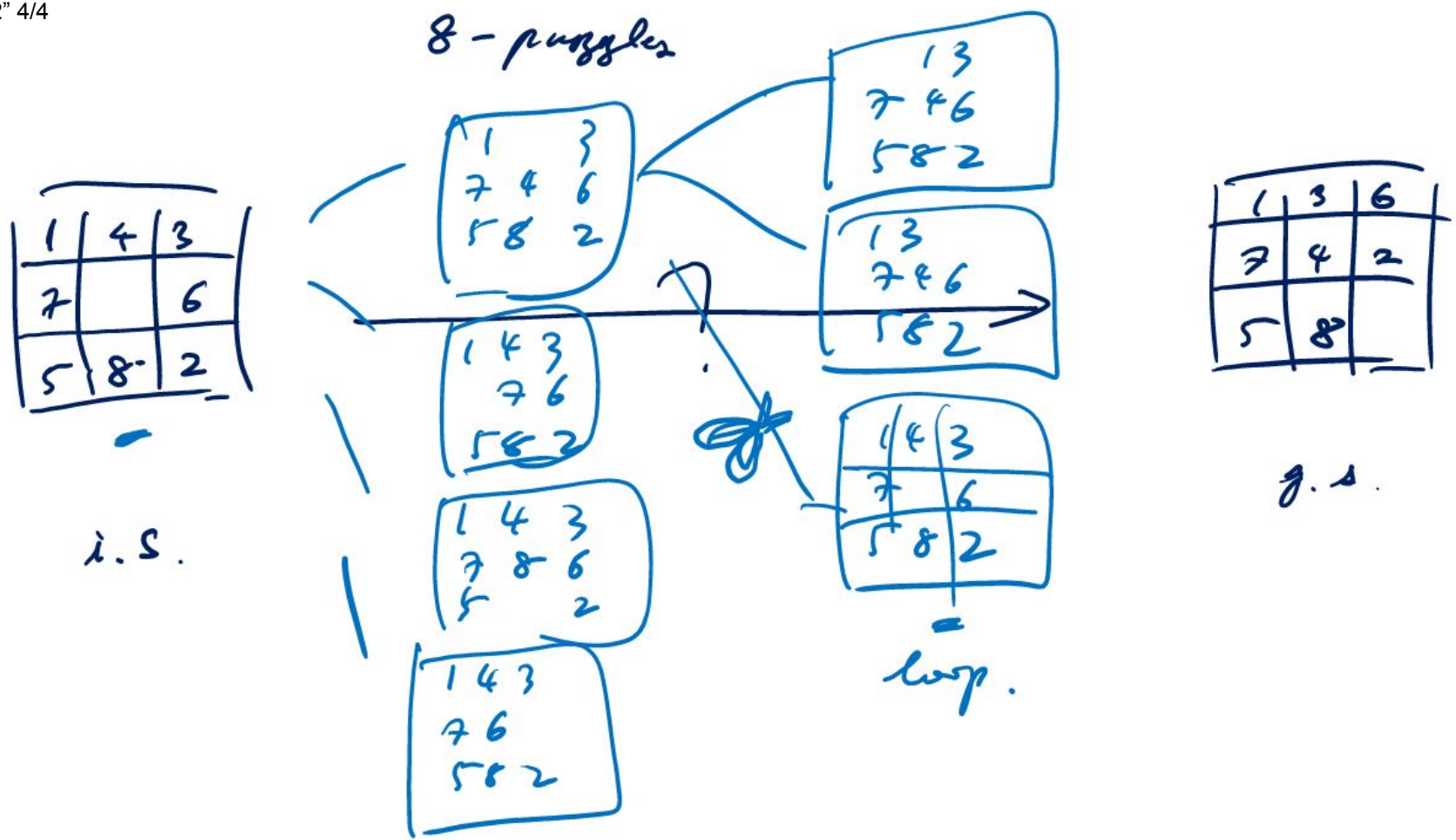
— yeah!!! (4 steps)

non-easy. why?

Tower of Hanoi

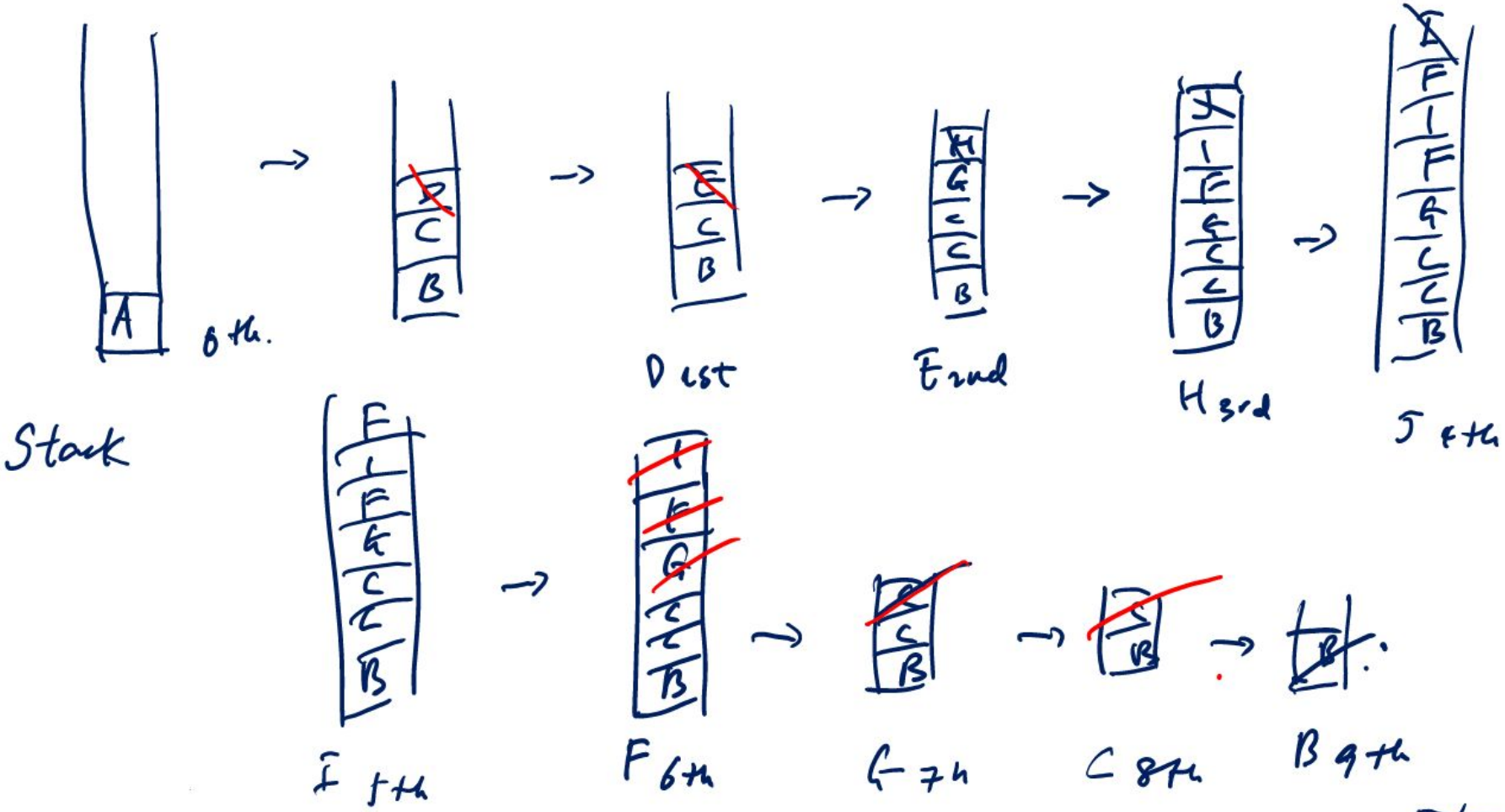


8 - puzzles



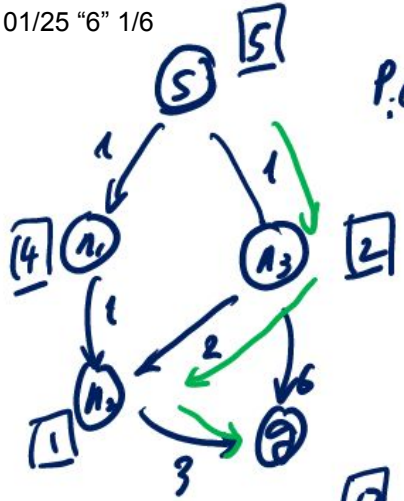
W3c4 lecture notes. Jan 18th

Graph Searching: DFS (Stack), LCFS.


A
0th.
D
C
B
D 1st
E
C
B
E 2nd
H
G
E
C
C
B
H 3rd
J
I
F
G
C
C
B
J 4th
Stack
F
I
E
G
C
C
B
I 5th
I
E
G
C
C
B
F 6th
C
B
G 7th
B
C 8th
B 9th

w3 c6 lecture notes. Jan 25

Admissibility and optimality of A* search



P.Q.

5.
|S|

$\{S\}$.

3	5
n_3	n_1

$\{S, n_3\}$

4	5	7 = 1+6+0
n_2'	n_1	g''

$\{S, n_2, n_3\}$

5	6	7
n_1	g'	g''

$\{S, n_1, n_2, n_3\}$

3	6	7
n_2	g'	g''

5	6	7
g	g'	g''

$S \to n_1 \to n_2 - g$

cost = 1+1+3 = 5

optimal.

6.	7
g'	g''

$S \to n_3 \to n_2 \to g$

Cost = 1+2+3 = 6.

Non-optimal.

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

Defn. (Admissibility of A^*). The A^* search is admissable <u>iff</u>, as long as solutions exist, an optimal solution (i.e., a shortest path from a start node s to the goal node g), will be found by A^*, ~~[illegible]~~ even if the search is infinite (i.e. graphs with cycles).

Thm. The search A^* is admissible if

— the branching factor of the 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.

pf: 1. we will first prove that A^* always find a soln. upon the settings above. If the search tree is with finite depth (no cycles), then the frontier F ~~with~~ will not be trapped into infinite cycles, and all nodes/paths will be explored sooner or later, including the soln paths. Since there exists at least one.

meanwhile. if the tree is infinite, it means some nodes will be inserted into F and selected from F repeatedly. But each time the node (say node n) is selected. The f-value will be increased,

$$f(n) = \underline{g(n)} + \underline{h(n)}.$$

Since the actual cost for reaching n from S, g(n) always increases. But this means a frontier node for the soln path will eventually be selected.

2.

2. we now prove that the 1st soln is optimal.

Suppose we have an optimal path in the form

$(S, n_1, n_2, \dots, n_k, g)$. $f(g) = g(g) \le g(g') = f(g')$

and $(S \to g')$ refers to some non-optimal soln.

we know that.

in other words $f(g') = g(g') + h(g') = g(g')$, $h(g') = 0$

$f(n_1) \le f(g) \le f(g')$

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

equals to the actual cost from S to g'.

Since n_1 is a neighbor of S. it must be inserted into F.

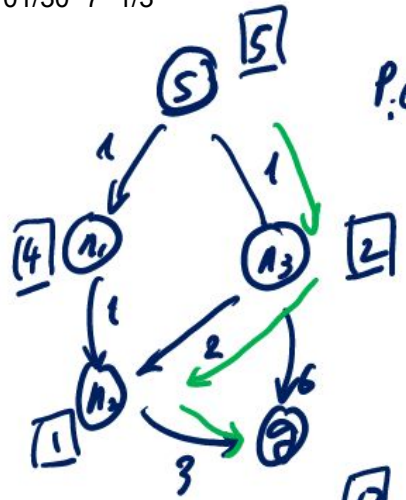
But. $f(n_1) = g(n_1) + h(n_1)$, is less than or equals to $f(g)$.

which is the minimal cost of the path to g from S.

$f(n_1) < f(g')$, which means n_1 will be expanded. ~~before~~ before g'. In addition, n_2 as a neighbour of n_1 will be inserted into F. 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/node g, will be selected before g'.

W4c7 lecture notes. Jan 30

monotone restriction for MPC (multiple-path checking) in A*, an exmaple.



P.Q.

5.
|S|

{S}.

3	5
n3	n1

{S, n3}

4	5	7 = 1+6+0
n2'	n1	g''

{S, n2, n3}

5	6	7
n1	g'	g''

{S, n1, n2, n3}

3	6	7
n2	g'	g''

5	6	7
g	g'	g''

S → n1 → n2 → g

cost = 1+1+3 = 5

optimal.

6.	7
g'	g''

S → n3 → n2 → g

Cost = 1+2+3 = 6.

Non-optimal.

p ← true.

little guy.

$KB \vdash \alpha$

muskIsRich

$(p \vee q)$

$\Updownarrow$

$(p \wedge q)$

$-(p \wedge \neg p$

$\mathcal{M} \models \alpha$

T T

fact

$(p \vee \neg p)$

KB → FOL

meta

ISO 9000

MPC

PQ

3

|s|

$s \to n_1 \to n_2 \to g$

{s}

3 3

$|n_1|n_3|$

{s, n_1}

$3 = 1+1+1$

3

$|n_2|n_3|$

{s, n_1, n_2}

3 $5 = 1+1+3+0$

$|n_3|g|$

MPC

{s, n_1, n_2, n_3}

5 7

$|\cancel{n_2}|g|g''|$

$|g''|$

{s, n_1, n_2, n_3, g}

W4c8 lecture notes. Feb 01

FOL via an example of Alice, Bob, Carol, and David.

alphabet/grammar FOL.

Constant: alice, bob, carol, david.

1-arity function: bestFriendOf.

0-arity predicate: ThisIsAliceWorld.

propositional.

1-arity predicate: Happy, Male, Female.

2-arity predicate: Couple. denotation.

$\forall$, $\exists$, $\wedge$ $\vee$ $\neg$

Interpretation I_1, denoted by terms.

$D = \{\triangle, \square, \triangleleft, \triangleright\}$ objects

Happy $P_1 = \{\triangle\}$

male $P_2 = \{\square, \triangleright\}$

Female $P_3 = \{\triangle, \triangleleft\}$

Couple $P_4 = \{(\triangle, \triangleright), (\triangleright, \triangle), (\triangleleft, \square), (\square, \triangleleft)\}$.

bestFriendOf $\left\{ \begin{array}{l} \triangleleft \to \triangle \\ \triangle \to \triangleleft \\ \square \to \square \\ \triangleright \to \square \end{array} \right\}$

sentences:

① Happy(alice)

② Happy(bob)

$\neg$ Happy(bob)

Happy(bestFriendOf(alice))

Happy(bestFriendOf(carol))

Couple(alice, bob)

Couple(alice, david)

Couple(carol, bestFriendOf(bob))

$\forall x$. Happy(x)

$\exists x$. Happy(x)

S_{all} S

$I_1 \models S$

$\dfrac{S \cup ①}{S_1}$

$I_1 \not\models S_1$

① $\vee$ ② S_i I_i

$I_i \models$ Happy(alice)

$S_i \not\models$

$\exists x$. Happy(x).