Week 6: Delete Relaxation

COMP90054 – AI Planning for Autonomy

Key concepts

- Delete relaxation heuristic h^+
- The relationship between h^{max}, h^{add} and h^+

Problem 1

What is the (optimal) delete relaxation heuristic h^+?

Relaxing by **ignoring delete lists**

Definition (Delete Relaxation).

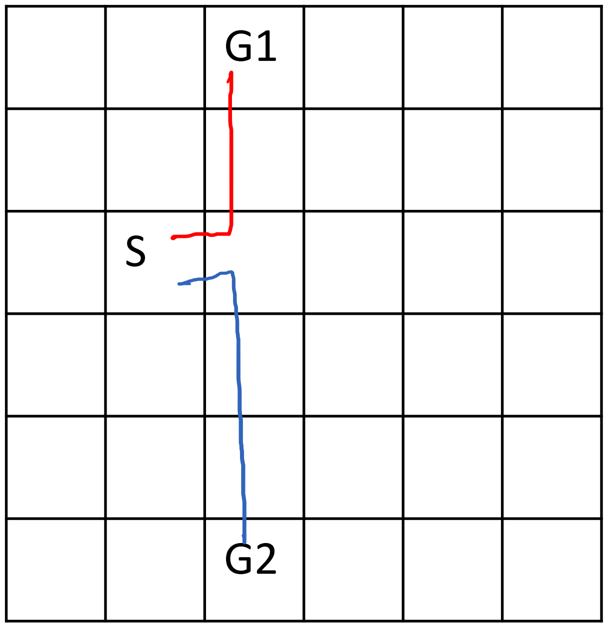
(i) *For a STRIPS action a, by a^+ we denote the corresponding delete relaxed action, or short relaxed action*, defined by $pre_{a^+} := pre_a$, $add_{a^+} := add_a$, and $del_{a^+} :=$

P = <F, O, I, G>

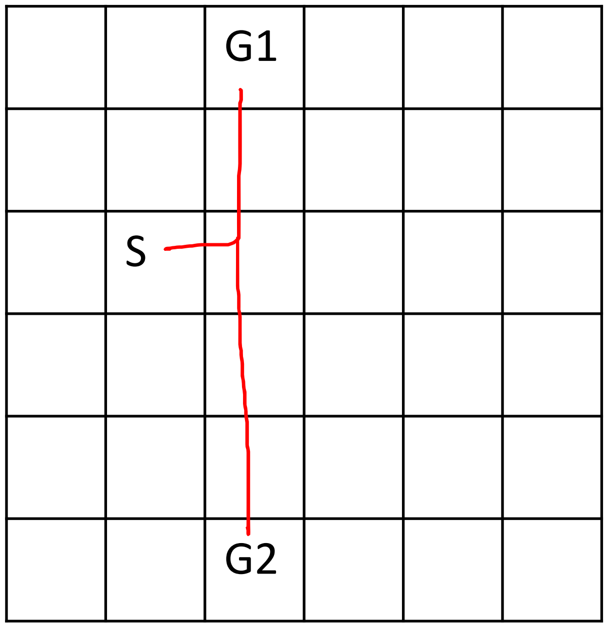
P = <F, O^+, I, G>

Problem 1

How would it be interpreted in pacman?



Minimum spanning tree:
Admissible, Not consistent



Minimum Steiner tree:
Admissible, consistent

Problem 1

What is the relationship between $\boldsymbol{h^{max}}, \boldsymbol{h^{add}}$ and $\boldsymbol{h^{+}}$? What about $\boldsymbol{h^{*}}$?

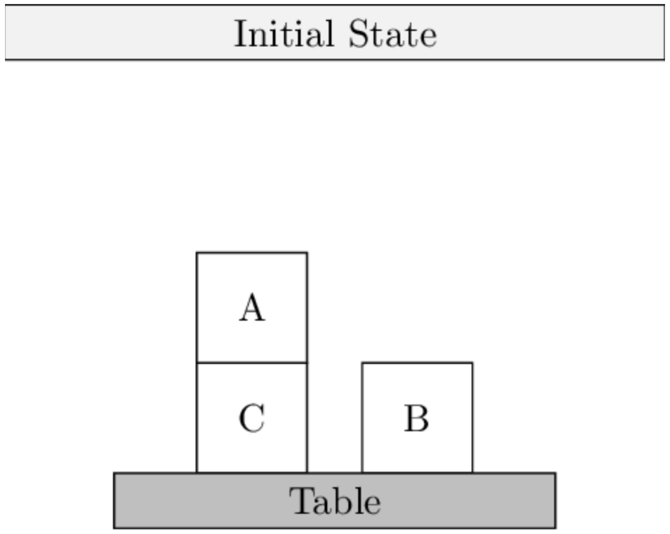
h* is the perfect heuristic (the optimal cost from the current state to the goal state)

h^{+} is the **optimal delete relaxation** heuristic (not easy to compute)
h^{+} is admissible

$h^{\boldsymbol{max}}$ is an approximation of h^{+}
$h^{\boldsymbol{max}}$ is admissible. $h^{\boldsymbol{max}}$ is very small.
$h^{\boldsymbol{max}}$ <= h^{+} <= h*

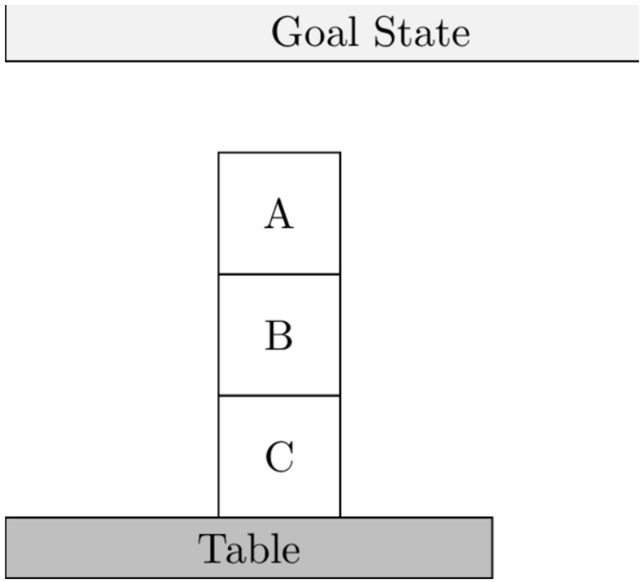
$h^{\boldsymbol{add}}$ is an approximation of h^{+}
$h^{\boldsymbol{add}}$ is not admissible
$h^{\boldsymbol{add}}$ >= h^{+}

Problem 2: Computing h^{max} and h^{add}



Initial state
I = {on(A, C), onTable(C), onTable(B), clear(A), clear(B), handFree}

Goal state
G = {on(A,B), on(B,C), onTable(C)}



Problem 2: Computing h^{max} and h^{add}

Definition (h^{add}). *Let $\Pi = (F, A, c, I, G)$ be a STRIPS planning task. The additive heuristic h^{add} for Π is the function $h^{\text{add}}(s) := h^{\text{add}}(s, G)$ where $h^{\text{add}}(s, g)$ is the point-wise greatest function that satisfies $h^{\text{add}}(s, g) =$*

$$\begin{cases} 0 & g \subseteq s \\ min_{a \in A, g \in add_a} c(a) + h^{\text{add}}(s, pre_a) & |g| = 1 \\ \sum_{g' \in g} h^{\text{add}}(s, \{g'\}) & |g| > 1 \end{cases}$$

Definition (h^{max}). *Let $\Pi = (F, A, c, I, G)$ be a STRIPS planning task. The max heuristic h^{max} for Π is the function $h^{\text{max}}(s) := h^{\text{max}}(s, G)$ where $h^{\text{max}}(s, g)$ is the point-wise greatest function that satisfies $h^{\text{max}}(s, g) =$*

$$\begin{cases} 0 & g \subseteq s \\ min_{a \in A, g \in add_a} c(a) + h^{\text{max}}(s, pre_a) & |g| = 1 \\ \max_{g' \in g} h^{\text{max}}(s, \{g'\}) & |g| > 1 \end{cases}$$

Problem 2: Computing h^{max} and h^{add}

**This table is not complete. Please refer to the solution notebook to see the complete table.*

Iter	c(A)	c(B)	c(C)	onT(A)	onT(B)	onT(C)	on(A,B)	on(A,C)	on(B,C)	hold(A)	hold(B)	hold(C)	handFree
0	0	0	∞	∞	0	0	∞	0	∞	∞	∞	∞	0
1													
2													

I = {on(A, C), onTable(C), onTable(B), clear(A), clear(B), handFree}

c(A) = clear(A)
onTable(A) = onT(A)
hold(A) = holding(A)

Problem 2: Computing h^{max} and h^{add}

Iter	c(A)	c(B)	c(C)	onT(A)	onT(B)	onT(C)	on(A,B)	on(A,C)	on(B,C)	hold(A)	hold(B)	hold(C)	handFree
0	0	0	∞	∞	0	0	∞	0	∞	∞	∞	∞	0
1	0	0	?		0	0		0					0
2													

Which actions can we take to make **clear(C**) True?

Problem 2

Which actions can we take to make **clear(C**) True?

putdown(C)
stack(C, A)
stack(C, B)
unstack(A, C)
unstack(B, C)

**Using the complete table, we need to add 2 more actions here: stack(C, C), unstack(C, C)*

Define Operators

O = {

pickup(x)
- Prec: onTable(x), clear(x), handFree
- Add: holding(x)
- Del: onTable(x), clear(x), handFree

unstack(x, y)
- Prec: on(x, y), clear(x), handFree
- Add: holding(x), clear(y)
- Del: on(x, y), clear(x), handFree

putdown(x)
- Prec: holding(x)
- Add: clear(x), onTable(x), handFree
- Del: holding(x)

stack(x, y)
- Prec: holding(x), clear(y)
- Add: clear(x), on(x,y), handFree
- Del: clear(y), holding(x)

}

Problem 2

Iter	c(A)	c(B)	c(C)	onT(A)	onT(B)	onT(C)	on(A,B)	on(A,C)	on(B,C)	hold(A)	hold(B)	hold(C)	handFree
0	0	0	∞	∞	0	0	∞	0	∞	∞	∞	∞	0
1	0	0	?		0	0		0					0
2													

h^{add} = action cost + ***sum***(heuristic of preconditions)

h^{max} = action cost + ***max***(heuristic of preconditions)

putdown(C) = 1 + hold(C) = 1 + ∞ = ∞

1 + hold(C) = ∞

stack(C, A) = 1 + hold(C) + clear(A) = 1 + ∞ + 0 = ∞

1 + max(hold(C), clear(A)) = 1 + ∞ = ∞

stack(C, B) = 1 + hold(C) + clear(B) = 1 + ∞ + 0 = ∞

1 + max(hold(C), clear(B)) = 1 + ∞ = ∞

unstack(A, C) = 1 + on(A, C) + clear(A) + handFree = 1 + 0 + 0 + 0 = 1

1 + max(on(A, C), clear(A), handFree) = 1

unstack(B, C) = 1 + on(B, C) + clear(B) + handFree = 1 + ∞ + 0 + 0 = ∞

1 + max(on(B, C), clear(B), handFree) = ∞

unstack(x, y)

- Prec: on(x, y), clear(x), handFree
- Add: holding(x), clear(y)
- Del: on(x, y), clear(x), handFree

putdown(x)

- Prec: holding(x)
- Add: clear(x), onTable(x), handFree
- Del: holding(x)

stack(x, y)

- Prec: holding(x), clear(y)
- Add: clear(x), on(x,y), handFree
- Del: clear(y), holding(x)

Problem 2

Iter	c(A)	c(B)	c(C)	onT(A)	onT(B)	onT(C)	on(A,B)	on(A,C)	on(B,C)	hold(A)	hold(B)	hold(C)	handFree
0	0	0	∞	∞	0	0	∞	0	∞	∞	∞	∞	0
1	0	0	?		0	0		0					0
2													

putdown(C) = ∞
stack(C, A) = ∞
stack(C, B) = ∞
unstack(A, C) = 1
unstack(B, C) = ∞

min(putdown(C), stack(C, A), stack(C, B), unstack(A, C), unstack(B, C)) = 1

Problem 2: Computing h^{max} and h^{add}

Iter	c(A)	c(B)	c(C)	onT(A)	onT(B)	onT(C)	on(A,B)	on(A,C)	on(B,C)	hold(A)	hold(B)	hold(C)	handFree
0	0	0	∞	∞	0	0	∞	0	∞	∞	∞	∞	0
1	0	0	1	∞	0	0	∞	0	∞	1	1	∞	0
2									?				

pickup(x)
- Prec: onTable(x), clear(x), handFree
- Add: holding(x)
- Del: onTable(x), clear(x), handFree

unstack(x, y)
- Prec: on(x, y), clear(x), handFree
- Add: holding(x), clear(y)
- Del: on(x, y), clear(x), handFree

putdown(x)
- Prec: holding(x)
- Add: clear(x), onTable(x), handFree
- Del: holding(x)

stack(x, y)
- Prec: holding(x), clear(y)
- Add: clear(x), on(x,y), handFree
- Del: clear(y), holding(x)

Problem 2: Computing h^{max} and h^{add}

Iter	c(A)	c(B)	c(C)	onT(A)	onT(B)	onT(C)	on(A,B)	on(A,C)	on(B,C)	hold(A)	hold(B)	hold(C)	handFree
0	0	0	∞	∞	0	0	∞	0	∞	∞	∞	∞	0
1	0	0	1	∞	0	0	∞	0	∞	1	1	∞	0
2									?				

h^{add} = action cost + ***sum***(heuristic of preconditions)

h^{max} = action cost + ***max***(heuristic of preconditions)

stack(B,C) = 1 + hold(B) + c(C) = 1 + 1 + 1 = 3

stack(B,C) = 1 + max(hold(B), c(C)) = 1 + 1 = 2

stack(x, y)

- Prec: holding(x), clear(y)
- Add: clear(x), on(x,y), handFree
- Del: clear(y), holding(x)

Problem 2: Computing h^{max} and h^{add}

Iter	c(A)	c(B)	c(C)	onT(A)	onT(B)	onT(C)	on(A,B)	on(A,C)	on(B,C)	hold(A)	hold(B)	hold(C)	handFree
0	0	0	∞	∞	0	0	∞	0	∞	∞	∞	∞	0
1	0	0	1	∞	0	0	∞	0	∞	1	1	∞	0
2	0	0	1	2	0	0	2	0	3 / 2	1	1	2	0

h^{add} / h^{max}

Problem 2: Computing h^{max} and h^{add}

Iter	c(A)	c(B)	c(C)	onT(A)	onT(B)	onT(C)	on(A,B)	on(A,C)	on(B,C)	hold(A)	hold(B)	hold(C)	handFree
0	0	0	∞	∞	0	0	∞	0	∞	∞	∞	∞	0
1	0	0	1	∞	0	0	∞	0	∞	1	1	∞	0
2	0	0	1	2	0	0	2	0	3 / 2	1	1	2	0
3	0	0	1	2	0	0	2	0	3 / 2	1	1	2	0

h^{add} / h^{max}

stop when converge (2 rows have the same values)

Problem 2: Computing h^{max} and h^{add}

Iter	c(A)	c(B)	c(C)	onT(A)	onT(B)	onT(C)	on(A,B)	on(A,C)	on(B,C)	hold(A)	hold(B)	hold(C)	handFree
0	0	0	∞	∞	0	0	∞	0	∞	∞	∞	∞	0
1	0	0	1	∞	0	0	∞	0	∞	1	1	∞	0
2	0	0	1	2	0	0	2	0	3 / 2	1	1	2	0
3	0	0	1	2	0	0	2	0	3 / 2	1	1	2	0

h^{add} / h^{max}

h^{add}(s0) = 2 + 3 + 0 = 5
$h^{max}(s0) = \max(2, 2, 0) = 2$

G = {on(A,B), on(B,C), onTable(C)}