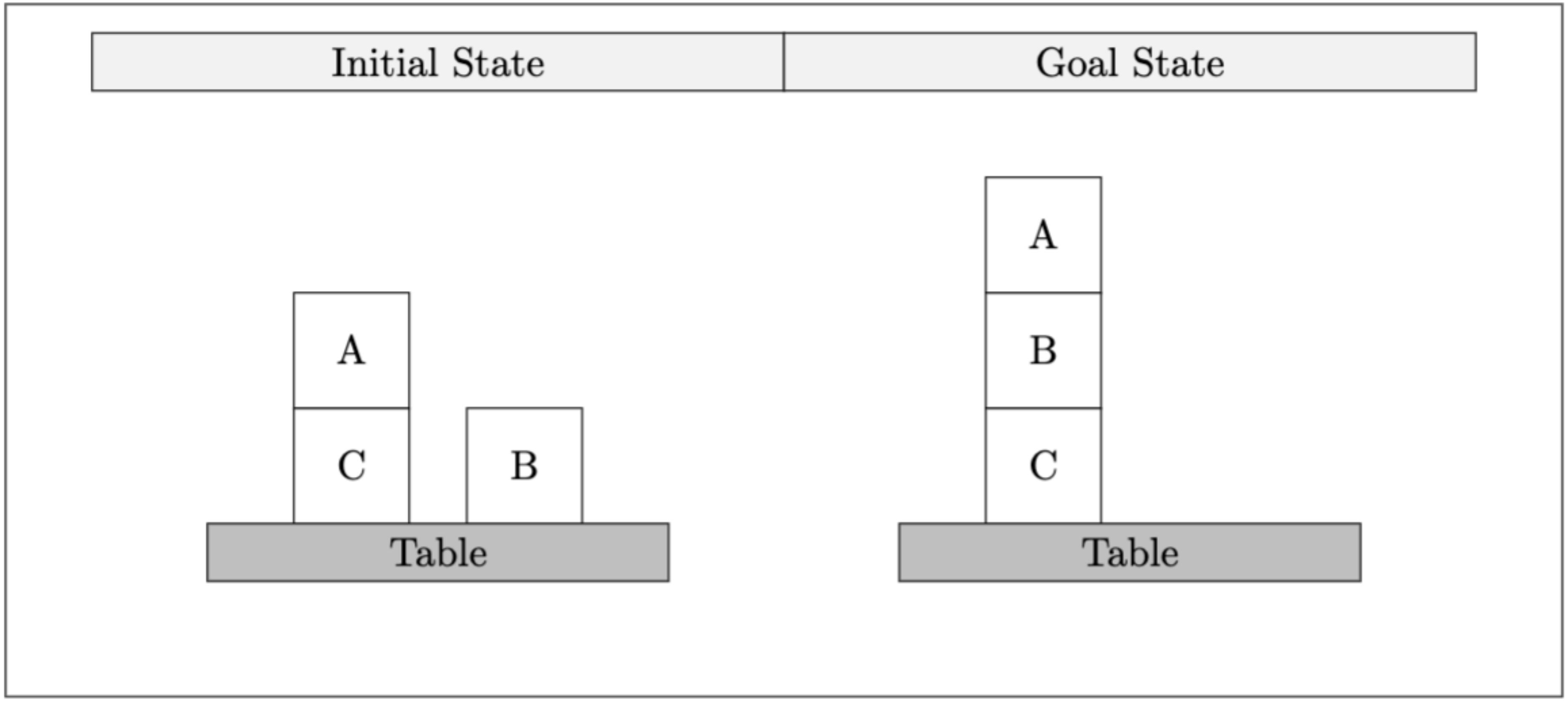
Workshop 6

Problem 1



Compute the values of each of the following heuristics for this problem

- h^{ff}: Use h^{max} for the best-supporters function.
- h^{ff}: Use h^{add} for the best-supporters function.

Recap

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 $h^{\max}(s) := h^{\max}(s, G)$ where $h^{\max}(s, g)$ is the point-wise greatest function that satisfies $h^{\max}(s, g) =$*

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

How to calculate hff (if using h_max)?

- Get the complete h_max table
- Follow the pseudo code to come up with a relaxed plan
- Sum up the cost of all actions (in the relaxed plan)

Relaxed Plan Extraction for state s and best-supporter function bs

$Open := G \setminus s$; $Closed := \emptyset$; $RPlan := \emptyset$
while $Open \neq \emptyset$ **do**:
  select $g \in Open$
  $Open := Open \setminus \{g\}$; $Closed := Closed \cup \{g\}$;
  $RPlan := RPlan \cup \{bs(g)\}$; $Open := Open \cup (pre_{bs(g)} \setminus (s \cup Closed))$
endwhile
return $RPlan$

The additive and max heuristics

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 $h^{\mathsf{add}}(s) := h^{\mathsf{add}}(s, G)$ where $h^{\mathsf{add}}(s, g)$ is the point-wise greatest function that satisfies $h^{\mathsf{add}}(s, g) =$*

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

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 $h^{\mathsf{max}}(s) := h^{\mathsf{max}}(s, G)$ where $h^{\mathsf{max}}(s, g)$ is the point-wise greatest function that satisfies $h^{\mathsf{max}}(s, g) =$*

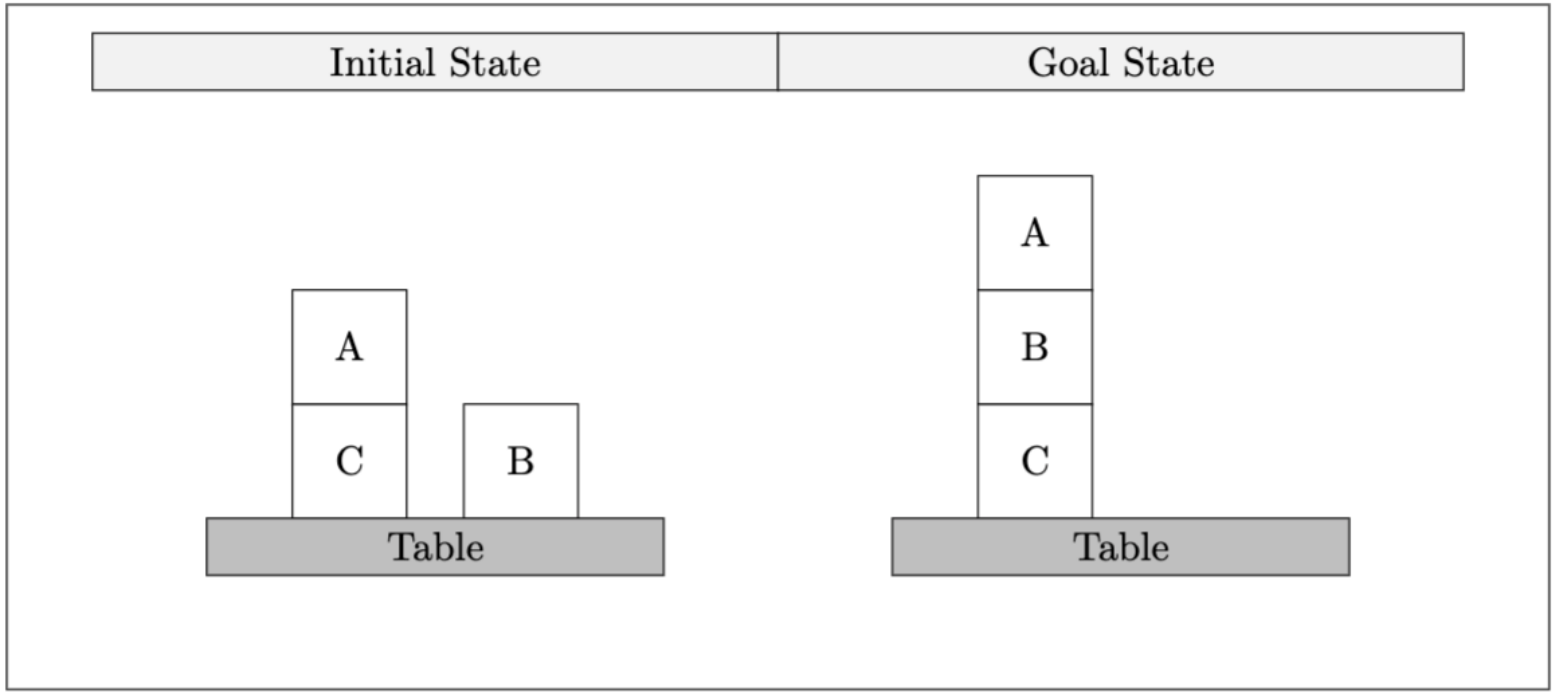
$$\begin{cases} 0 & g \subseteq s \\ min_{a \in A, g \in add_a} c(a) + h^{\mathsf{max}}(s, pre_a) & |g| = 1 \\ \max_{g' \in g} h^{\mathsf{max}}(s, \{g'\}) & |g| > 1 \end{cases}$$

Best Support Functions

Definition (Best-Supporters from $h^{\max}$ and h^{add}). *Let* $\Pi = (F, A, c, I, G)$ *be a STRIPS planning task, and let* s *be a state.*
The $h^{\max}$ supporter function $bs_s^{\max} : \{p \in F \mid 0 < h^{\max}(s, \{p\}) < \infty\} \mapsto A$ *is defined by* $bs_s^{\max}(p) := \arg\min_{a \in A, p \in add_a} c(a) + h^{\max}(s, pre_a)$.
The h^{add} supporter function $bs_s^{\text{add}} : \{p \in F \mid 0 < h^{\text{add}}(s, \{p\}) < \infty\} \mapsto A$ *is defined by* $bs_s^{\text{add}}(p) := \arg\min_{a \in A, p \in add_a} c(a) + h^{\text{add}}(s, pre_a)$.

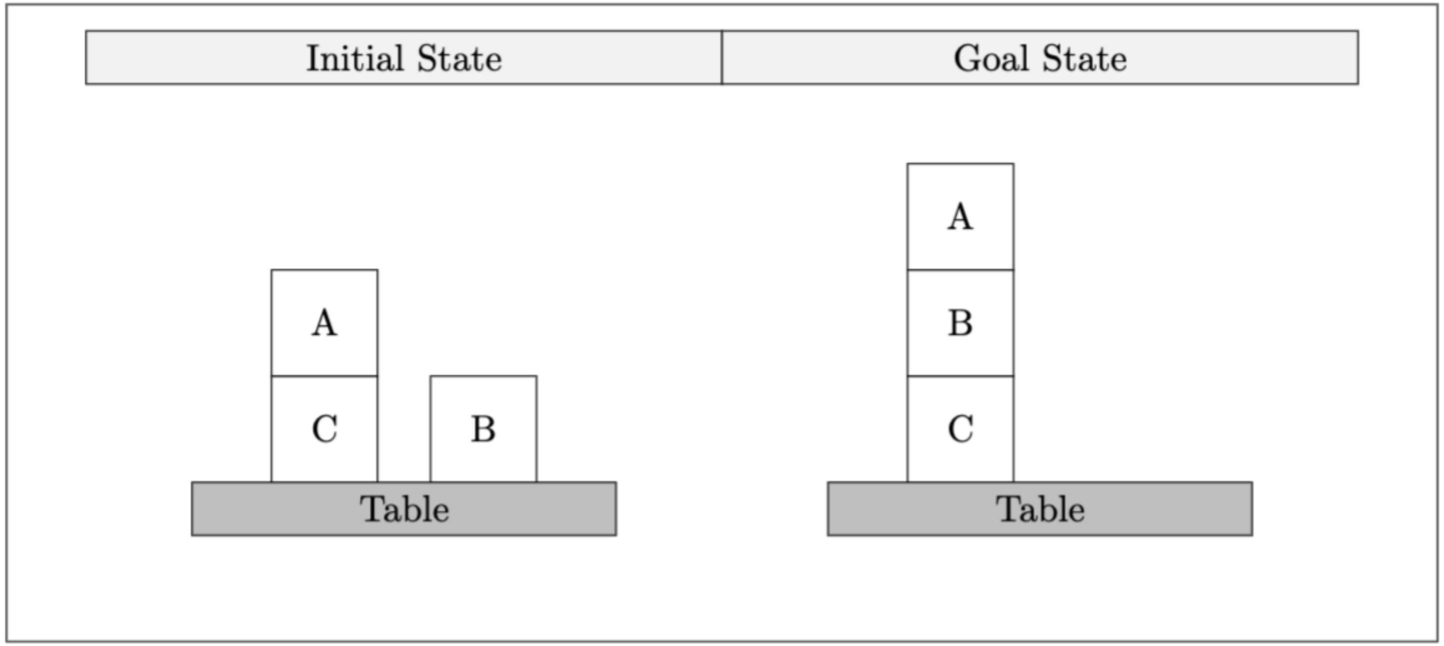
Problem 1

- Open List = {?}
- Closed List = {?}
- Relaxed Plan = {}



Problem 1

- Open List = {on(A, B), on(B, C)}
- Closed List = {on(A, C), onTable(C), onTable(B), clear(A), clear(B), armFree}
- Relaxed Plan = {}



Iteration 1

- Open List = {on(A, B), on(B, C)}
- Closed List = {on(A, C), onTable(C), onTable(B), clear(A), clear(B), armFree}
- Relaxed Plan = {}

- **Select on(A,B) from the open list**
- what are the actions can make on(A,B) true? Only stack(A,B)
- bs(on(A,B)) = 1 + h_max(holding(A), clear(B))) = 1 + max(1,0) = 2

- Open List = {on(B, C)} U {holding(A)}
- Closed List = Closed List U {on(A, B)}
- Relaxed Plan = Relaxed Plan U {stack(A,B)}

Iteration 2

- Open List = {holding(A), on(B, C)}
- **Select holding(A) from the open list**
- what are the actions can make holding(A) true?
- bs(holding(A))=
- Pickup(A): = 1 + h_max(c(A), aF, onT(A)) = 1 + max(0,0,2) = 3
- Unstack(A,A) = 1 + h_max(c(A), aF, on(A, A)) = 1 + max(0,0,2) = 3
- Unstack(A,B) = 1 + h_max(c(A), aF, on(A, B)) = 1 + max(0,0,2) = 3
- Unstack(A,C) = 1 + h_max(c(A), aF, on(A, C) = 1 + max(0,0,0) = 1

- Open List = {on(B,C)} U {empty list}
- Closed List = Closed List U {holding(A)}
- Relaxed Plan = Relaxed Plan U {Unstack(A,C)}

Iteration 3

- Open List = {on(B, C)}
- Relaxed Plan = {stack(A,B), unstack(A,C)}

- **Select on(B, C) from the open list**
- what are the actions can make on(B, C) true? Only stack(B, C)
- bs(on(B, C)) = 1 + h_max(holding(B), clear(C))) = 1 + max(1,0) = 2

- Open List = {} U {holding(B), clear(C)}
- Closed List = Closed List U {on(B, C)}
- Relaxed Plan = Relaxed Plan U {stack(B, C)}

Iteration 4

- Open List = {holding(B), clear(C)}
- **Select holding(B) from the open list**
- what are the actions can make holding(B) true?
- bs(holding(B))=
- Pickup(B): = 1 + h_max(c(A), aF, onT(A)) = 1 + max(0,0,0) = 1
- Unstack(B,A)
- Unstack(B,B)
- Unstack(B,C)

- Open List = {clear(C)}U {empty list}
- Closed List = Closed List U {holding(B)}
- Relaxed Plan = Relaxed Plan U {Pickup(B)}

Iteration 5

- Open List = {clear(C)}
- Relaxed Plan = {stack(A,B), unstack(A,C), stack(B, C), pickup(B)}
- **Select clear(C) from the open list**
- what are the actions can make clear(C) true?
- Putdown(C), Stack(C,A), Stack(C,B), Stack(A,C), Stack(B,C), Stack(B,C), Stack(C,C)
- Unstack(A,C)= 1 + h_max(c(A), aF, on(A, C)) = 1+ max(0,0,0) = 1

- Relaxed Plan = ?
- hff = ?

Consider the following problem: $G = \{A, B, C\}$, $I = \{\}, O = \{add(o_a) = \{A\}, add(o_b) = \{B\}, add(o_c) = \{C\}\}$, which following option is correct?

- $h^* = 3, h_{max} = 1,\ h_{add} = 1$
- $h^* = 3, h_{max} = 1,\ h_{add} = 3$
- $h^* = 3, h_{max} = 3,\ h_{add} = 1$
- $h^* = 3, h_{max} = 3,\ h_{add} = 3$

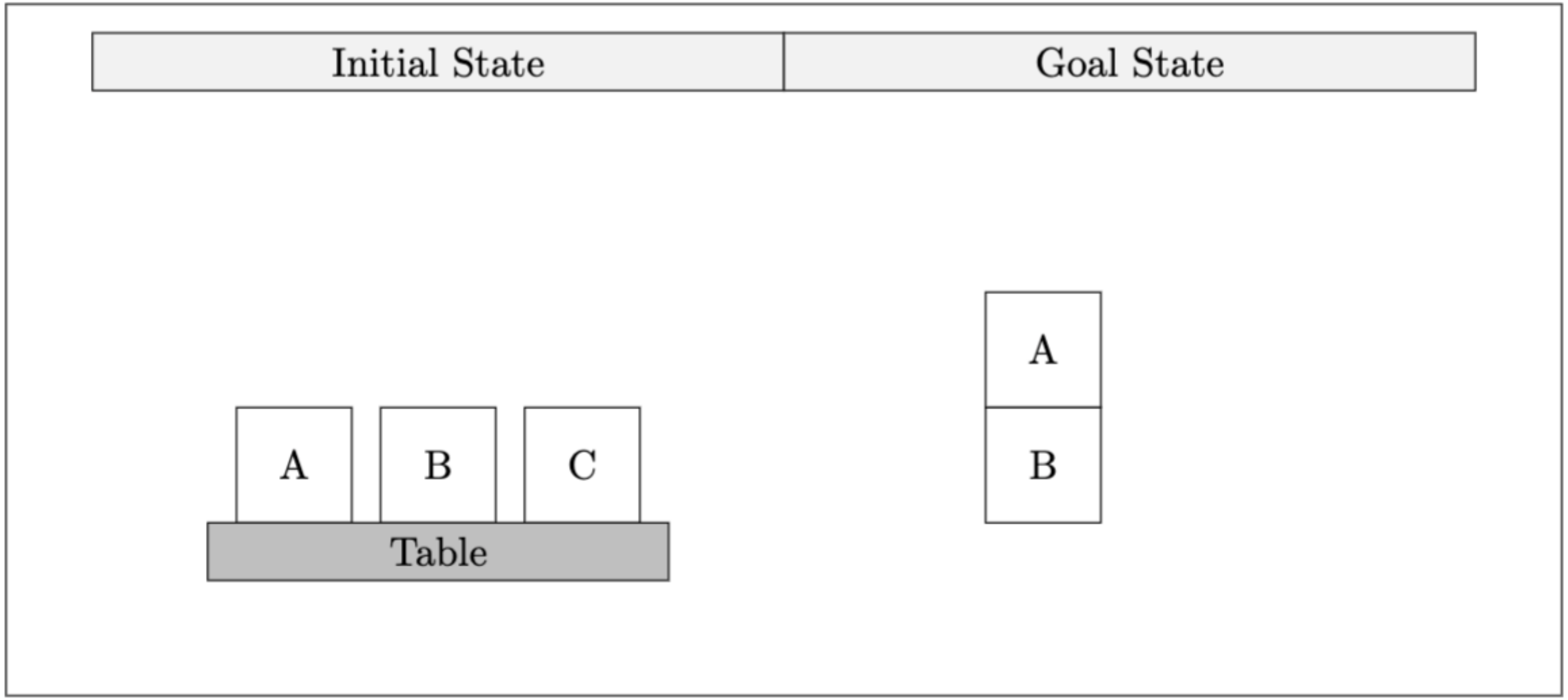
- Relaxed Plan = {Oa} U {Ob} U {Oc} = {Oa, Ob, Oc}
- hff = 3

Consider the following problem: $G = \{A, B, C\}$, $I = \{\}, O = \{add(o_a) = \{A, B, C\}\}$, which following option is correct?

- $h^* = 1, h_{max} = 1,\ h_{add} = 1$
- $h^* = 1, h_{max} = 1,\ h_{add} = 3$
- $h^* = 1, h_{max} = 3,\ h_{add} = 1$
- $h^* = 1, h_{max} = 3,\ h_{add} = 3$

- Relaxed Plan = {Oa} U {Oa} U {Oa} = {Oa}
- hff = 1

Problem 2 Part 1



- Show the IW(1) search tree for this problem, highlighting each state why it passes the novelty pruning test or why is being pruned. IW(1) should solve this problem. Stop as soon as you find a state the satisfies the goal condition.

IW vs ID

- **Iterative Deepening (ID): DFS with depth limit l**
- l starts from 0, l = 0, l = 1……l = d
- **Iterated Width(IW): BFS with width limit k**
- k starts from 0, k = 0, k = 1……until we find a solution

Key definition: the **novelty** $w(s)$ **of a state** s is the size of the smallest subset of atoms in s that is true for the first time in the search.

- e.g. $w(s) = 1$ if there is **one** atom $p \in s$ such that s is the first state that makes p true.
- Otherwise, $w(s) = 2$ if there are **two** different atoms $p, q \in s$ such that s is the first state that makes $p \wedge q$ true.

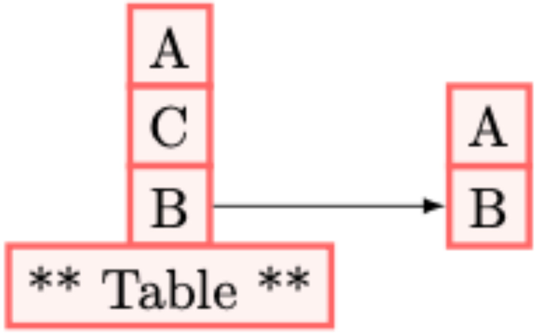
IW(k) = breadth-first search that prunes newly generated states whose $novelty(s) > k$.

Problem 2 Part 2

- Can you think of an initial situation where IW(1) cannot find a solution for the goal on(A,B), but IW(2) does, explain your answer?

Initial State: { on(A,C), On(C,B), OnTable(B), armEmpty }

Goal: { on(A,B) }



Problem 2 Part 2

- Can you think of an initial situation where IW(1) cannot find a solution for the goal on(A,B), but IW(2) does, explain your answer?

Operators (Stack & Unstack)

O:=

{ stack(x,y): =

prec:= {holding(x), clear(y)}

add:= {clear(x), on(x,y), armFree}

del:= {clear(y), holding(x)}

| x, y ∈ {A, B, C} and $x \neq y$ }

}

^ {unstack(x,y): =

prec:= {on(x,y), clear(x), armFree}

add:= {holding(x), clear(y)}

del:= {clear(x), on(x,y), armFree}

| x, y ∈ {A, B, C} and $x \neq y$ }

}

Operators (putdown & pickup)

^ { putdown(x): =

    prec:= {holding(x) }

    add:= {clear(x), onTable(x), armFree}

    del:= {holding(x)}

    | x, y ∈ {A, B, C}}

}

^ {pickup(x): =

    prec:= {onTable(x), clear(x), armFree}

    add:= {holding(x)}

    del:= {clear(x), onTable(x), armFree}

    | x, y ∈ {A, B, C}}

}

Problem 2 Part 2

- Can you think of an initial situation where IW(1) cannot find a solution for the goal on(A,B), but IW(2) does, explain your answer?

Initial State: { on(A,C), On(C,B), OnTable(B), armEmpty }

Goal: { on(A,B) }