

Week 7: Relaxed Plan Heuristics and Iterated Width (IW)

COMP90054 – AI Planning for Autonomy

Key concepts

- Relaxed Plan Heuristics (h^{ff})
- Iterated Width (IW)

Relaxed Plan Heuristics (h^{ff})

- h^* is the perfect heuristic
- h^+ is the **optimal delete relaxation** heuristic (not easy to compute)
- h^{max} is an approximation of h^+
- h^{add} is an approximation of h^+
- h^{ff} is an approximation of h^+

	Pros	Cons
h^{max}	Admissible	Very small (optimistic)
h^{add}	More informed than h^{max}	Not admissible (pessimistic) over-counting

h^{ff} can reduce over-counting (but it is still inadmissible)

Find h^{ff} based on h^{max} and h^{add}

Problem 1: Computing h^{ff}

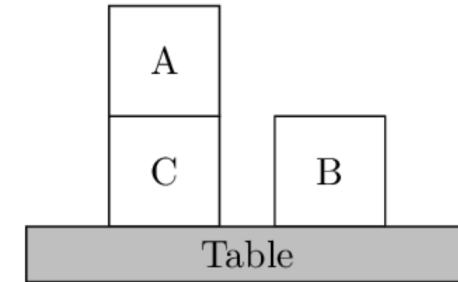
Initial state

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

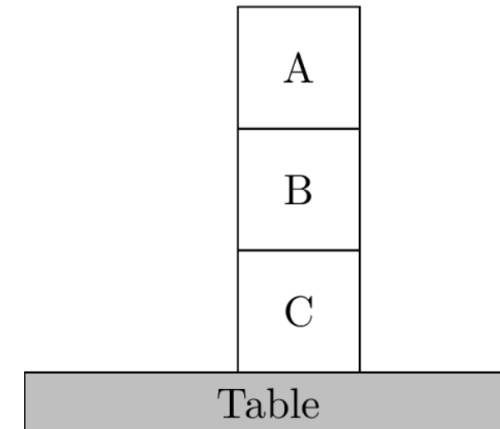
Goal state

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

Initial State



Goal State

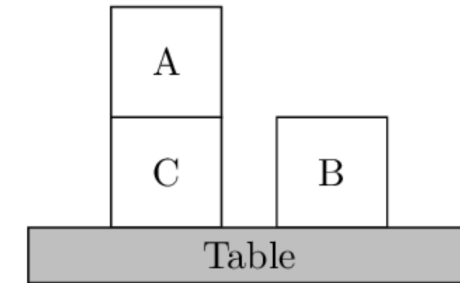


Problem 1: Computing h^{ff}

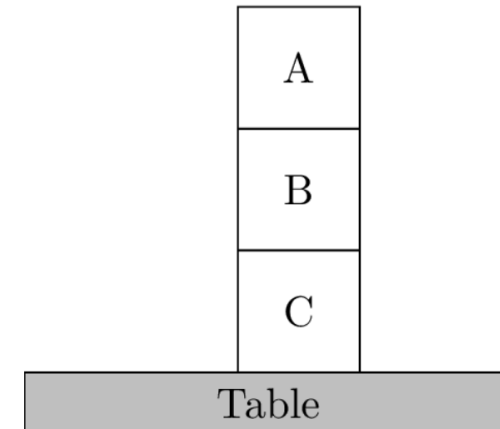
1. Find best-supporter function (bs)

2. Relaxed Plan Extraction for state s

Initial State



Goal State



Problem 1: Find the best-supporter function for each fact

h^{add} / h^{max}

Iter	c(A)	c(B)	c(C)	hand Free	h(A)	h(B)	h(C)	on(A, A)	on(A,B)	on(A,C)	on(B,A)	on(B,B)	on(B,C)	on(C,A)	on(C,B)	on(C,C)	onT(A)	onT(B)	onT(C)
0	0	0	∞	0	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
1	0	0	1	0	1	1	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
2	0	0	1	0	1	1	2	2	2	0	2	2	3 / 2	∞	∞	∞	2	0	0
3	0	0	1	0	1	1	2	2	2	0	2	2	3 / 2	3	3	4 / 3	2	0	0
4	0	0	1	0	1	1	2	2	2	0	2	2	3 / 2	3	3	4 / 3	2	0	0

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

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

2. Which action is the best-supporter function of **clear(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 1: Find the best-supporter function for each fact

h^{add} / h^{max}

Iter	c(A)	c(B)	c(C)	hand Free	h(A)	h(B)	h(C)	on(A, A)	on(A,B)	on(A,C)	on(B,A)	on(B,B)	on(B,C)	on(C,A)	on(C,B)	on(C,C)	onT(A)	onT(B)	onT(C)
0	0	0	∞	0	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
1	0	0	1	0	1	1	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
2	0	0	1	0	1	1	2	2	2	0	2	2	3 / 2	∞	∞	∞	2	0	0
3	0	0	1	0	1	1	2	2	2	0	2	2	3 / 2	3	3	4 / 3	2	0	0
4	0	0	1	0	1	1	2	2	2	0	2	2	3 / 2	3	3	4 / 3	2	0	0

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)

2. Which action is the best-supporter function of **clear(C)**?

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

Use h^{add}

$$\text{putdown(C)} = 1 + \text{hold(C)} = 1 + 2 = 3$$

$$\text{stack(C, A)} = 1 + \text{hold(C)} + \text{clear(A)} = 1 + 2 + 0 = 3$$

$$\text{stack(C, B)} = 1 + \text{hold(C)} + \text{clear(B)} = 1 + 2 + 0 = 3$$

$$\text{stack(C, C)} = 1 + \text{hold(C)} + \text{clear(C)} = 1 + 2 + 1 = 4$$

$$\text{unstack(A, C)} = 1 + \text{on(A, C)} + \text{clear(A)} + \text{handFree} = 1 + 0 + 0 + 0 = 1$$

$$\text{unstack(B, C)} = 1 + \text{on(B, C)} + \text{clear(B)} + \text{handFree} = 1 + 3 + 0 + 0 = 4$$

$$\text{unstack(C, C)} = 1 + \text{on(C, C)} + \text{clear(C)} + \text{handFree} = 1 + 4 + 1 + 0 = 6$$

Problem 1: Find the best-supporter function for each fact

h^{add} / h^{max}

Iter	c(A)	c(B)	c(C)	hand Free	h(A)	h(B)	h(C)	on(A, A)	on(A,B)	on(A,C)	on(B,A)	on(B,B)	on(B,C)	on(C,A)	on(C,B)	on(C,C)	onT(A)	onT(B)	onT(C)
0	0	0	∞	0	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
1	0	0	1	0	1	1	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
2	0	0	1	0	1	1	2	2	2	0	2	2	3 / 2	∞	∞	∞	2	0	0
3	0	0	1	0	1	1	2	2	2	0	2	2	3 / 2	3	3	4 / 3	2	0	0
4	0	0	1	0	1	1	2	2	2	0	2	2	3 / 2	3	3	4 / 3	2	0	0

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)

2. Which action is the best-supporter function of **clear(C)**?

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

Use h^{max}

$$\text{putdown(C)} = 1 + \text{hold(C)} = 1 + 2 = 3$$

$$\text{stack(C, A)} = 1 + \max(\text{hold(C)}, \text{clear(A)}) = 1 + \max(2, 0) = 3$$

$$\text{stack(C, B)} = 1 + \max(\text{hold(C)}, \text{clear(B)}) = 1 + \max(2, 0) = 3$$

$$\text{stack(C, C)} = 1 + \max(\text{hold(C)}, \text{clear(C)}) = 1 + \max(2, 1) = 3$$

$$\text{unstack(A, C)} = 1 + \max(\text{on(A, C)}, \text{clear(A)}, \text{handFree}) = 1 + \max(0, 0, 0) = 1$$

$$\text{unstack(B, C)} = 1 + \max(\text{on(B, C)}, \text{clear(B)}, \text{handFree}) = 1 + \max(2, 0, 0) = 3$$

$$\text{unstack(C, C)} = 1 + \max(\text{on(C, C)}, \text{clear(C)}, \text{handFree}) = 1 + \max(3, 1, 0) = 4$$

Problem 1: Find the best-supporter function for each fact

h^{add} / h^{max}

Iter	c(A)	c(B)	c(C)	hand Free	h(A)	h(B)	h(C)	on(A, A)	on(A,B)	on(A,C)	on(B,A)	on(B,B)	on(B,C)	on(C,A)	on(C,B)	on(C,C)	onT(A)	onT(B)	onT(C)
0	0	0	∞	0	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
1	0	0	1	0	1	1	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
2	0	0	1	0	1	1	2	2	2	0	2	2	3/2	∞	∞	∞	2	0	0
3	0	0	1	0	1	1	2	2	2	0	2	2	3/2	3	3	4/3	2	0	0
4	0	0	1	0	1	1	2	2	2	0	2	2	3/2	3	3	4/3	2	0	0

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)

stack(B, C)

2. Which action is the best-supporter function of **on(B, C)**?

Problem 1: Find the best-supporter function for each fact

h^{add} / h^{max}

Iter	c(A)	c(B)	c(C)	hand Free	h(A)	h(B)	h(C)	on(A, A)	on(A,B)	on(A,C)	on(B,A)	on(B,B)	on(B,C)	on(C,A)	on(C,B)	on(C,C)	onT(A)	onT(B)	onT(C)
0	0	0	∞	0	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
1	0	0	1	0	1	1	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞	0	0
2	0	0	1	0	1	1	2	2	2	0	2	2	$3 / 2$	∞	∞	∞	2	0	0
3	0	0	1	0	1	1	2	2	2	0	2	2	$3 / 2$	3	3	$4 / 3$	2	0	0
4	0	0	1	0	1	1	2	2	2	0	2	2	$3 / 2$	3	3	$4 / 3$	2	0	0

Use h^{add} / h^{max} for the best-supporter function

	clear(a)	clear(b)	clear(c)	handempty()	holding(a)	holding(b)	holding(c)	on(a,a)	on(a,b)	on(a,c)	on(b,a)	on(b,b)	on(b,c)	on(c,a)	on(c,b)	on(c,c)	ontable(a)	ontable(b)	ontable(c)
0	NA	NA	(unstack a c)	NA	(unstack a c)	(pick-up b)	(pick-up c)	(stack a a)	(stack a b)	NA	(stack b a)	(stack b b)	(stack b c)	(stack c a)	(stack c b)	(stack c c)	(put-down a)	NA	NA

Problem 1: Relaxed Plan Extraction

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

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

	clear(a)	clear(b)	clear(c)	handempty()	holding(a)	holding(b)	holding(c)	on(a,a)	on(a,b)	on(a,c)	on(b,a)	on(b,b)	on(b,c)	on(c,a)	on(c,b)	on(c,c)	ontable(a)	ontable(b)	ontable(c)
0	NA	NA	(unstack a c)	NA	(unstack a c)	(pick-up b)	(pick-up c)	(stack a a)	(stack a b)	NA	(stack b a)	(stack b b)	(stack b c)	(stack c a)	(stack c b)	(stack c c)	(put-down a)	NA	NA

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

```

Open := G \ s; Closed := ∅; RPlan := ∅
while Open ≠ ∅ do:
  select g ∈ Open
  Open := Open \ {g}; Closed := Closed ∪ {g};
  RPlan := RPlan ∪ {bs(g)}; Open := Open ∪ (prebs(g) \ (s ∪ Closed))
endwhile
return RPlan

```

$s = I$

$Open = \{\text{on}(A,B), \text{on}(B,C)\}$
 $Closed = \{\}$
 $RPlan = \{\}$

$G \setminus I$

Problem 1: Relaxed Plan Extraction

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

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

pickup(x)

- Prec: $\text{onTable}(x), \text{clear}(x), \text{handFree}$
- Add: $\text{holding}(x)$
- Del: $\text{onTable}(x), \text{clear}(x), \text{handFree}$

unstack(x, y)

- Prec: $\text{on}(x, y), \text{clear}(x), \text{handFree}$
- Add: $\text{holding}(x), \text{clear}(y)$
- Del: $\text{on}(x, y), \text{clear}(x), \text{handFree}$

putdown(x)

- Prec: $\text{holding}(x)$
- Add: $\text{clear}(x), \text{onTable}(x), \text{handFree}$
- Del: $\text{holding}(x)$

stack(x, y)

- Prec: $\text{holding}(x), \text{clear}(y)$
- Add: $\text{clear}(x), \text{on}(x,y), \text{handFree}$
- Del: $\text{clear}(y), \text{holding}(x)$

	clear(a)	clear(b)	clear(c)	handempty()	holding(a)	holding(b)	holding(c)	on(a,a)	on(a,b)	on(a,c)	on(b,a)	on(b,b)	on(b,c)	on(c,a)	on(c,b)	on(c,c)	ontable(a)	ontable(b)	ontable(c)
0	NA	NA	(unstack a c)	NA	(unstack a c)	(pick-up b)	(pick-up c)	(stack a a)	(stack a b)	NA	(stack b a)	(stack b b)	(stack b c)	(stack c a)	(stack c b)	(stack c c)	(put-down a)	NA	NA

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

```

Open := G \ s; Closed := ∅; RPlan := ∅
while Open ≠ ∅ do:
  select g ∈ Open
  Open := Open \ {g}; Closed := Closed ∪ {g};
  RPlan := RPlan ∪ {bs(g)}; Open := Open ∪ (prebs(g) \ (s ∪ Closed))
endwhile
return RPlan

```

$Open = \{\text{on}(A,B), \text{on}(B,C)\} \cup \{\text{holding}(A)\}$
 $Closed = \{\} \cup \{\text{on}(A,B)\}$
 $RPlan = \{\} \cup \{\text{stack}(A,B)\}$

Iteration 1:

- Select g from Open $g = \text{on}(A,B)$
 - Put g into Closed
 - Get $bs(g)$ and add $bs(g)$ into RPlan $bs(\text{on}(A,B)) = \text{stack}(A,B)$
 - Get $preconditions$ of $bs(g)$ and update Open list if necessary $pre_{\text{stack}(A,B)} = \{\text{holding}(A), \text{clear}(B)\}$
- $pre_{\text{stack}(A,B)} \setminus (I \cup Closed) = \{\text{holding}(A)\}$

Problem 1: Relaxed Plan Extraction

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

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

pickup(x)

- Prec: $\text{onTable}(x), \text{clear}(x), \text{handFree}$
- Add: $\text{holding}(x)$
- Del: $\text{onTable}(x), \text{clear}(x), \text{handFree}$

unstack(x, y)

- Prec: $\text{on}(x, y), \text{clear}(x), \text{handFree}$
- Add: $\text{holding}(x), \text{clear}(y)$
- Del: $\text{on}(x, y), \text{clear}(x), \text{handFree}$

putdown(x)

- Prec: $\text{holding}(x)$
- Add: $\text{clear}(x), \text{onTable}(x), \text{handFree}$
- Del: $\text{holding}(x)$

stack(x, y)

- Prec: $\text{holding}(x), \text{clear}(y)$
- Add: $\text{clear}(x), \text{on}(x,y), \text{handFree}$
- Del: $\text{clear}(y), \text{holding}(x)$

	clear(a)	clear(b)	clear(c)	handempty()	holding(a)	holding(b)	holding(c)	on(a,a)	on(a,b)	on(a,c)	on(b,a)	on(b,b)	on(b,c)	on(c,a)	on(c,b)	on(c,c)	ontable(a)	ontable(b)	ontable(c)
0	NA	NA	(unstack a c)	NA	(unstack a c)	(pick-up b)	(pick-up c)	(stack a a)	(stack a b)	NA	(stack b a)	(stack b b)	(stack b c)	(stack c a)	(stack c b)	(stack c c)	(put-down a)	NA	NA

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

```

Open := G \ s; Closed := ∅; RPlan := ∅
while Open ≠ ∅ do:
  select g ∈ Open
  Open := Open \ {g}; Closed := Closed ∪ {g};
  RPlan := RPlan ∪ {bs(g)}; Open := Open ∪ (prebs(g) \ (s ∪ Closed))
endwhile
return RPlan

```

$Open = \{\text{on}(B,C), \text{holding}(A)\} \cup \{\text{holding}(B), \text{clear}(C)\}$
 $Closed = \{\text{on}(A, B)\} \cup \{\text{on}(B,C)\}$
 $RPlan = \{\text{stack}(A, B)\} \cup \{\text{stack}(B,C)\}$

Iteration 2:

- Select g from Open $g = \text{on}(B,C)$

- Put g into Closed

- Get $bs(g)$ and add $bs(g)$ into RPlan $bs(\text{on}(B,C)) = \text{stack}(B,C)$

- Get $preconditions$ of $bs(g)$ and update Open list if necessary $pre_{\text{stack}(B,C)} = \{\text{holding}(B), \text{clear}(C)\}$

$pre_{\text{stack}(B,C)} \setminus (I \cup Closed) = \{\text{holding}(B), \text{clear}(C)\}$

Problem 1: Relaxed Plan Extraction

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

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

pickup(x)

- Prec: $\text{onTable}(x), \text{clear}(x), \text{handFree}$
- Add: $\text{holding}(x)$
- Del: $\text{onTable}(x), \text{clear}(x), \text{handFree}$

putdown(x)

- Prec: $\text{holding}(x)$
- Add: $\text{clear}(x), \text{onTable}(x), \text{handFree}$
- Del: $\text{holding}(x)$

unstack(x, y)

- Prec: $\text{on}(x, y), \text{clear}(x), \text{handFree}$
- Add: $\text{holding}(x), \text{clear}(y)$
- Del: $\text{on}(x, y), \text{clear}(x), \text{handFree}$

stack(x, y)

- Prec: $\text{holding}(x), \text{clear}(y)$
- Add: $\text{clear}(x), \text{on}(x,y), \text{handFree}$
- Del: $\text{clear}(y), \text{holding}(x)$

	clear(a)	clear(b)	clear(c)	handempty()	holding(a)	holding(b)	holding(c)	on(a,a)	on(a,b)	on(a,c)	on(b,a)	on(b,b)	on(b,c)	on(c,a)	on(c,b)	on(c,c)	ontable(a)	ontable(b)	ontable(c)
0	NA	NA	(unstack a c)	NA	(unstack a c)	(pick-up b)	(pick-up c)	(stack a a)	(stack a b)	NA	(stack b a)	(stack b b)	(stack b c)	(stack c a)	(stack c b)	(stack c c)	(put-down a)	NA	NA

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

```

Open := G \ s; Closed := ∅; RPlan := ∅
while Open ≠ ∅ do:
  select g ∈ Open
  Open := Open \ {g}; Closed := Closed ∪ {g};
  RPlan := RPlan ∪ {bs(g)}; Open := Open ∪ (prebs(g) \ (s ∪ Closed))
endwhile
return RPlan

```

$Open = \{\text{holding}(A), \text{holding}(B), \text{clear}(C)\}$

$Closed = \{\text{on}(A, B), \text{on}(B, C)\}$

$RPlan = \{\text{stack}(A, B), \text{stack}(B, C)\}$

$\cup \{\text{holding}(A)\}$
 $\cup \{\text{unstack}(A, C)\}$

Iteration 3:

- Select g from Open

$g = \text{holding}(A)$

- Put g into Closed

- Get $bs(g)$ and add $bs(g)$ into RPlan

$bs(\text{holding}(A)) = \text{unstack}(A, C)$

- Get *preconditions* of $bs(g)$ and update Open list if necessary

$\text{pre}_{\text{unstack}(A, C)} = \{\text{on}(A, C), \text{clear}(A), \text{handFree}\}$

$\text{pre}_{\text{unstack}(A, C)} \setminus (I \cup \text{Closed}) = \{\}$

Problem 1: Relaxed Plan Extraction

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

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

pickup(x)

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

putdown(x)

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

unstack(x, y)

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

stack(x, y)

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

	clear(a)	clear(b)	clear(c)	handempty()	holding(a)	holding(b)	holding(c)	on(a,a)	on(a,b)	on(a,c)	on(b,a)	on(b,b)	on(b,c)	on(c,a)	on(c,b)	on(c,c)	ontable(a)	ontable(b)	ontable(c)
0	NA	NA	(unstack a c)	NA	(unstack a c)	(pick-up b)	(pick-up c)	(stack a a)	(stack a b)	NA	(stack b a)	(stack b b)	(stack b c)	(stack c a)	(stack c b)	(stack c c)	(put-down a)	NA	NA

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

```

Open := G \ s; Closed := ∅; RPlan := ∅
while Open ≠ ∅ do:
  select g ∈ Open
  Open := Open \ {g}; Closed := Closed ∪ {g};
  RPlan := RPlan ∪ {bs(g)}; Open := Open ∪ (prebs(g) \ (s ∪ Closed))
endwhile
return RPlan

```

$Open = \{\text{holding}(B), \text{clear}(C)\}$

$Closed = \{\text{on}(A, B), \text{on}(B, C), \text{holding}(A)\}$

$RPlan = \{\text{stack}(A, B), \text{stack}(B, C), \text{unstack}(A, C)\}$

$\cup \{\text{holding}(B)\}$
 $\cup \{\text{pickup}(B)\}$

Iteration 4:

- Select g from Open

$g = \text{holding}(B)$

- Put g into Closed

- Get $bs(g)$ and add $bs(g)$ into RPlan

$bs(\text{holding}(B)) = \text{pickup}(B)$

- Get $preconditions$ of $bs(g)$ and update Open list if necessary

$pre_{\text{pickup}}(B) = \{\text{onTable}(B), \text{clear}(B), \text{handFree}\}$

$pre_{\text{pickup}}(B) \setminus (I \cup Closed) = \{\}$

Problem 1: Relaxed Plan Extraction

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

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

pickup(x)

- Prec: $\text{onTable}(x), \text{clear}(x), \text{handFree}$
- Add: $\text{holding}(x)$
- Del: $\text{onTable}(x), \text{clear}(x), \text{handFree}$

unstack(x, y)

- Prec: $\text{on}(x, y), \text{clear}(x), \text{handFree}$
- Add: $\text{holding}(x), \text{clear}(y)$
- Del: $\text{on}(x, y), \text{clear}(x), \text{handFree}$

putdown(x)

- Prec: $\text{holding}(x)$
- Add: $\text{clear}(x), \text{onTable}(x), \text{handFree}$
- Del: $\text{holding}(x)$

stack(x, y)

- Prec: $\text{holding}(x), \text{clear}(y)$
- Add: $\text{clear}(x), \text{on}(x, y), \text{handFree}$
- Del: $\text{clear}(y), \text{holding}(x)$

	clear(a)	clear(b)	clear(c)	handempty()	holding(a)	holding(b)	holding(c)	on(a,a)	on(a,b)	on(a,c)	on(b,a)	on(b,b)	on(b,c)	on(c,a)	on(c,b)	on(c,c)	ontable(a)	ontable(b)	ontable(c)
0	NA	NA	(unstack a c)	NA	(unstack a c)	(pick-up b)	(pick-up c)	(stack a a)	(stack a b)	NA	(stack b a)	(stack b b)	(stack b c)	(stack c a)	(stack c b)	(stack c c)	(put-down a)	NA	NA

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

```

Open := G \ s; Closed := ∅; RPlan := ∅
while Open ≠ ∅ do:
  select g ∈ Open
  Open := Open \ {g}; Closed := Closed ∪ {g};
  RPlan := RPlan ∪ {bs(g)}; Open := Open ∪ (prebs(g) \ (s ∪ Closed))
endwhile
return RPlan

```

$Open = \{\text{clear}(C)\}$

$Closed = \{\text{on}(A, B), \text{on}(B, C), \text{holding}(A), \text{holding}(B)\} \cup \{\text{clear}(C)\}$

$RPlan = \{\text{stack}(A, B), \text{stack}(B, C), \text{unstack}(A, C), \text{pickup}(B)\}$

Iteration 5:

- Select g from Open $g = \text{clear}(C)$

- Put g into Closed

- Get $bs(g)$ and add $bs(g)$ into RPlan $bs(\text{clear}(C)) = \text{unstack}(A, C)$

- Get $preconditions$ of $bs(g)$ and update Open list if necessary $pre_{\text{unstack}}(A, C) = \{\text{on}(A, C), \text{clear}(A), \text{handFree}\}$

$pre_{\text{unstack}}(A, C) \setminus (I \cup Closed) = \{\}$

\Rightarrow stop the iteration

Problem 1: Get h^{ff}

$RPlan = \{stack(A, B), stack(B, C), unstack(A, C), pickup(B)\}$

h^{ff} is the sum of the cost of actions in the relaxed plan

$h^{ff} = 4$ for both h^{max} and h^{add} (because they have the same best supporter functions for all facts)

Problem 2: Iterated Width (IW)

Iterated Width (IW) vs Iterative Deepening (ID)

- Both are blind search algorithms
- ID: DFS with depth limit
- IW: BFS with width limit

Problem 2: Iterated Width (IW)

Find the novelty $w(s)$ of a state s ?

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.
- Otherwise, $w(s) = 3$ if there are **three** different atoms...

Algorithm

- $IW(k)$ = **breadth-first** search that **prunes** newly generated states whose $\text{novelty}(s) > k$.
- IW is a **sequence of calls** $IW(k)$ for $i = 0, 1, 2, \dots$ over problem P until problem solved or i exceeds number of variables in problem

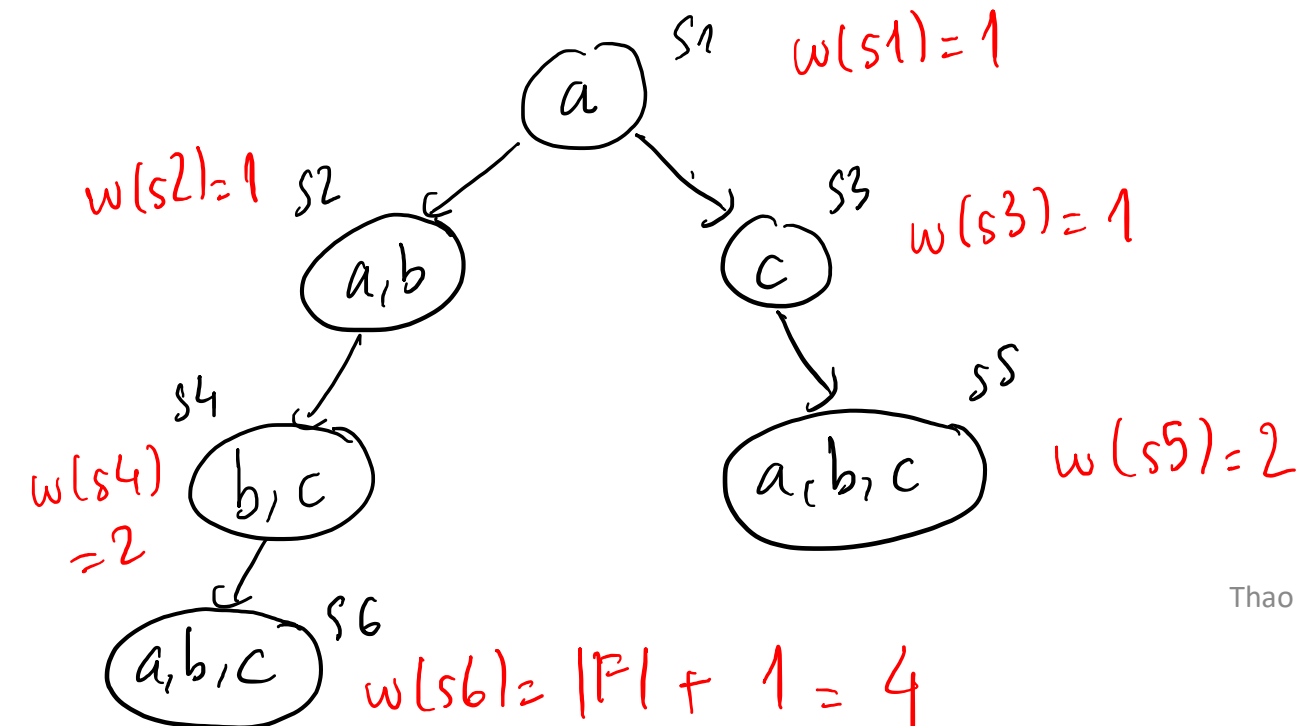
$$F = \{a, b, c\}$$

Queue

Novelty table

Have seen?

Novelty table	Have seen?
a	
b	
c	
a, b	
a, c	
b, c	
a, b, c	



Thao Le

Problem 2: Iterated Width (IW)

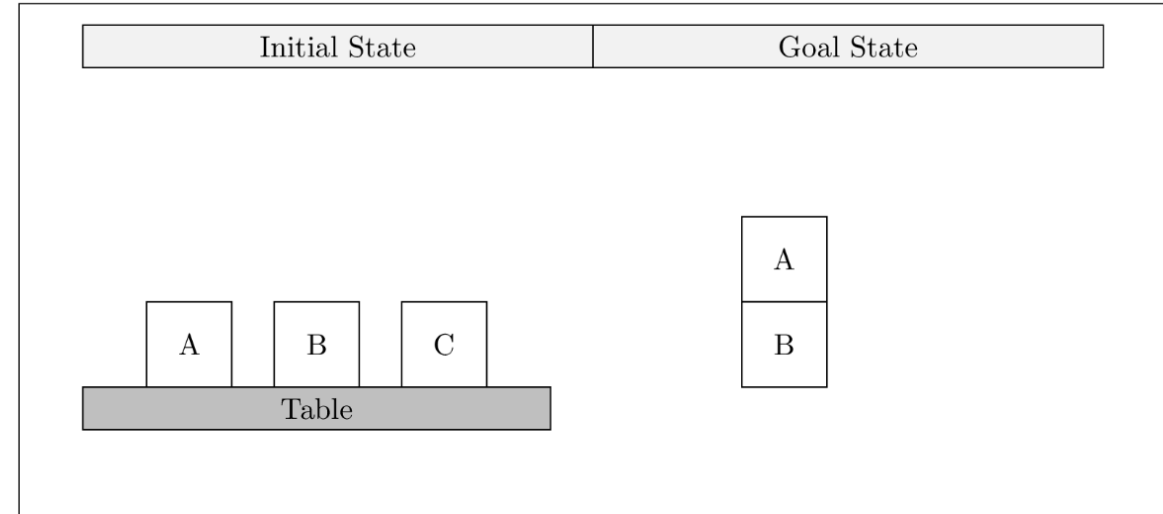
Show the IW(1): Prune when novelty(s) > 1

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

G = {on(A, B)}

F = {}

Novelty table



Problem 2: Iterated Width (IW)

Show the IW(1): Prune when novelty(s) > 1

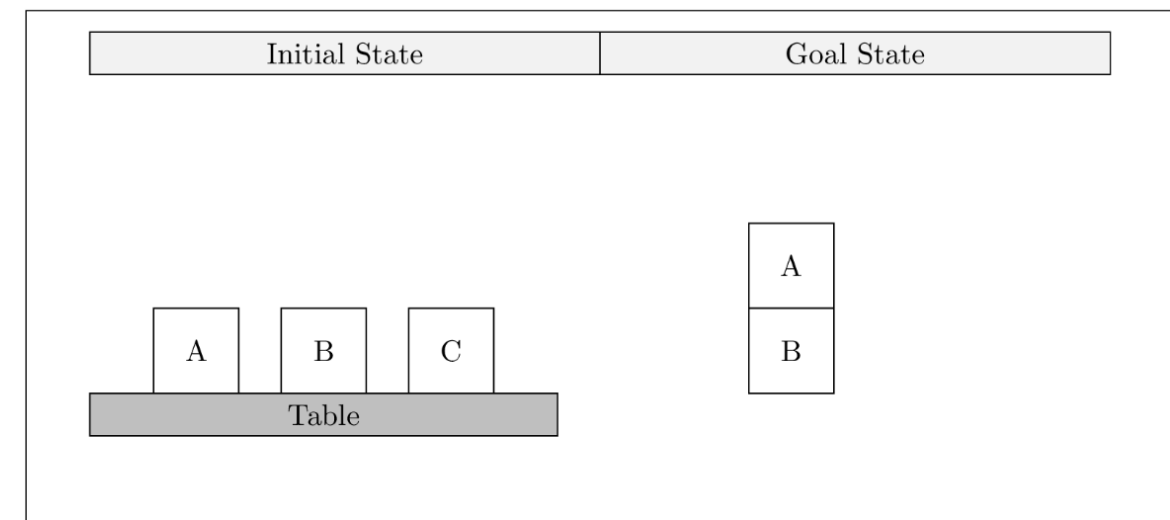
I = {onTable(A), onTable(B), onTable(C), clear(A), clear(B), clear(C), handFree}

G = {on(A, B)}



Novelty table:

onTable(A), onTable(B), onTable(C), clear(A), clear(B), clear(C), handFree

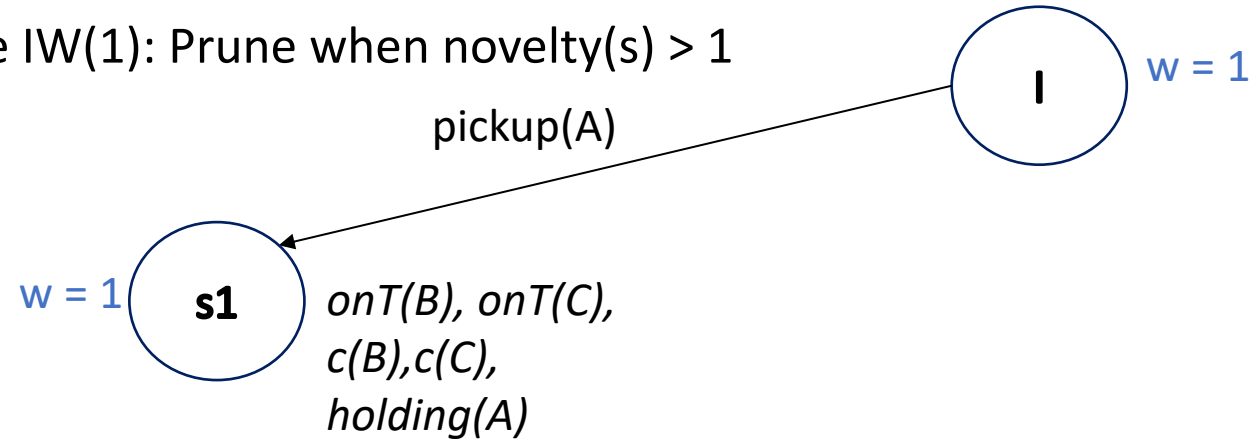


Problem 2: Iterated Width (IW)

$I = \{\text{onTable}(A), \text{onTable}(B), \text{onTable}(C), \text{clear}(A), \text{clear}(B), \text{clear}(C), \text{handFree}\}$

$G = \{\text{on}(A, B)\}$

Show the IW(1): Prune when novelty(s) > 1



Novelty table

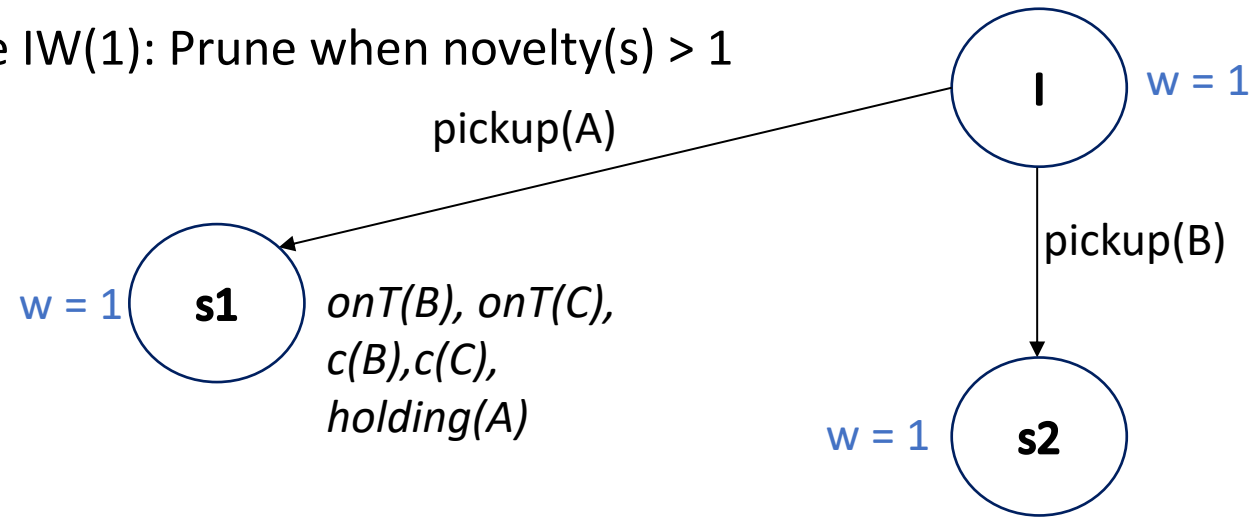
$\text{onTable}(A), \text{onTable}(B), \text{onTable}(C), \text{clear}(A), \text{clear}(B), \text{clear}(C), \text{handFree},$
 $\text{holding}(A)$

Problem 2: Iterated Width (IW)

$I = \{\text{onTable}(A), \text{onTable}(B), \text{onTable}(C), \text{clear}(A), \text{clear}(B), \text{clear}(C), \text{handFree}\}$

$G = \{\text{on}(A, B)\}$

Show the IW(1): Prune when novelty(s) > 1



Novelty table

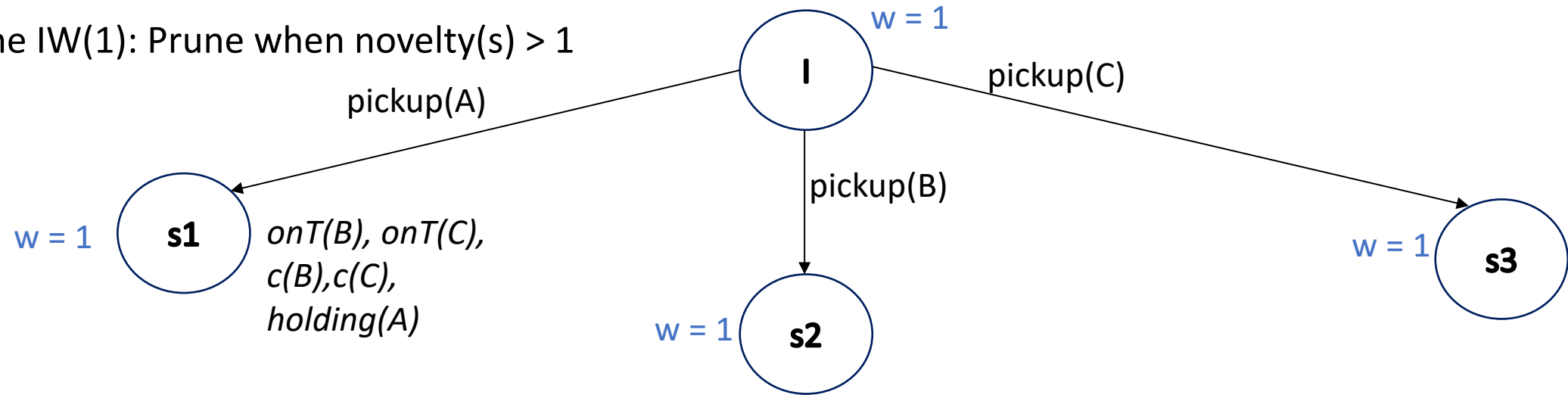
`onTable(A), onTable(B), onTable(C), clear(A), clear(B), clear(C), handFree,`
`holding(A), holding(B)`

Problem 2: Iterated Width (IW)

$I = \{\text{onTable}(A), \text{onTable}(B), \text{onTable}(C), \text{clear}(A), \text{clear}(B), \text{clear}(C), \text{handFree}\}$

$G = \{\text{on}(A, B)\}$

Show the IW(1): Prune when novelty(s) > 1



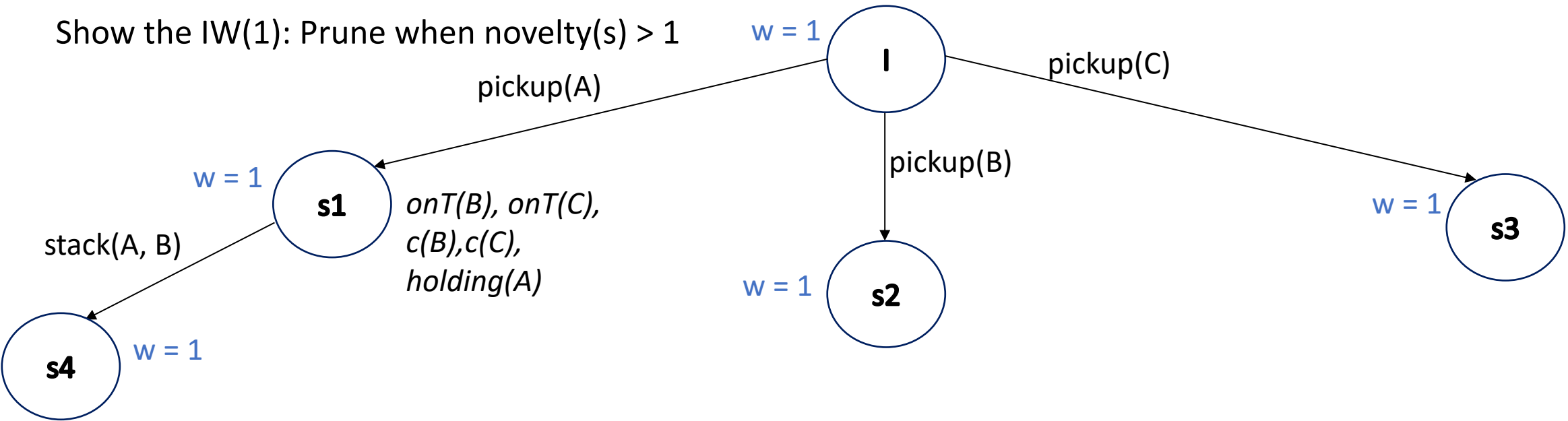
Novelty table

onTable(A), onTable(B), onTable(C), clear(A), clear(B), clear(C), handFree,
holding(A), holding(B), holding(C)

Problem 2: Iterated Width (IW)

$I = \{\text{onTable}(A), \text{onTable}(B), \text{onTable}(C), \text{clear}(A), \text{clear}(B), \text{clear}(C), \text{handFree}\}$
 $G = \{\text{on}(A, B)\}$

Show the IW(1): Prune when novelty(s) > 1



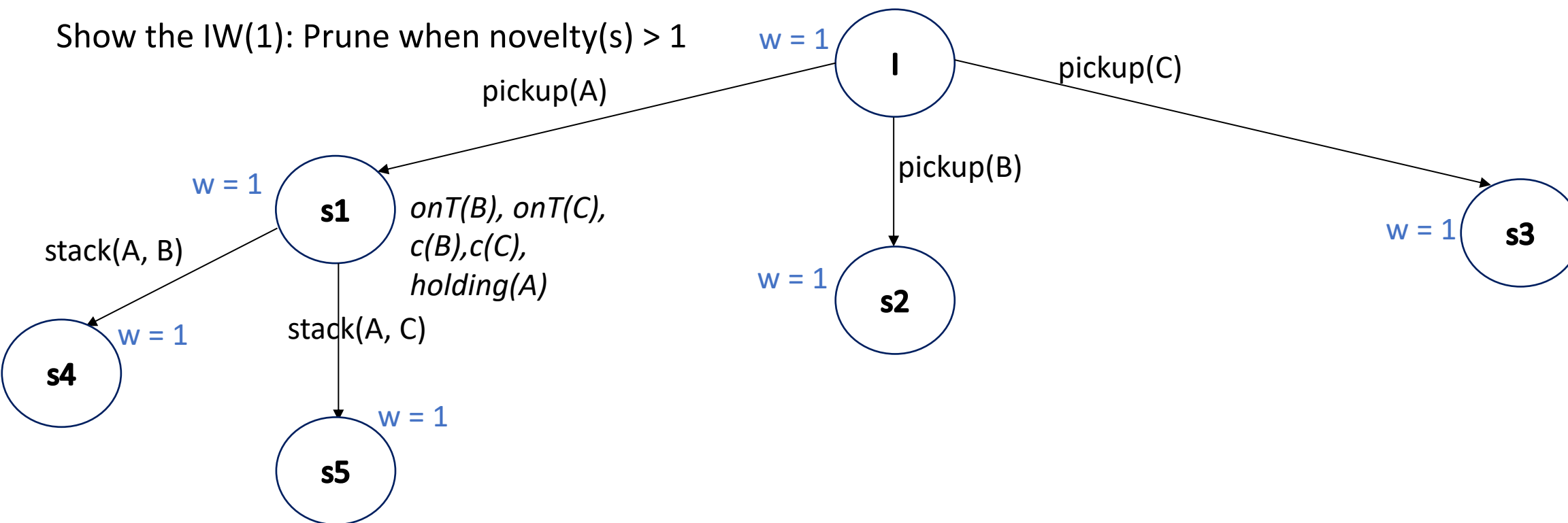
Novelty table

onTable(A), onTable(B), onTable(C), clear(A), clear(B), clear(C), handFree,
holding(A), holding(B), holding(C),
on(A, B)

Problem 2: Iterated Width (IW)

$I = \{\text{onTable}(A), \text{onTable}(B), \text{onTable}(C), \text{clear}(A), \text{clear}(B), \text{clear}(C), \text{handFree}\}$
 $G = \{\text{on}(A, B)\}$

Show the IW(1): Prune when novelty(s) > 1



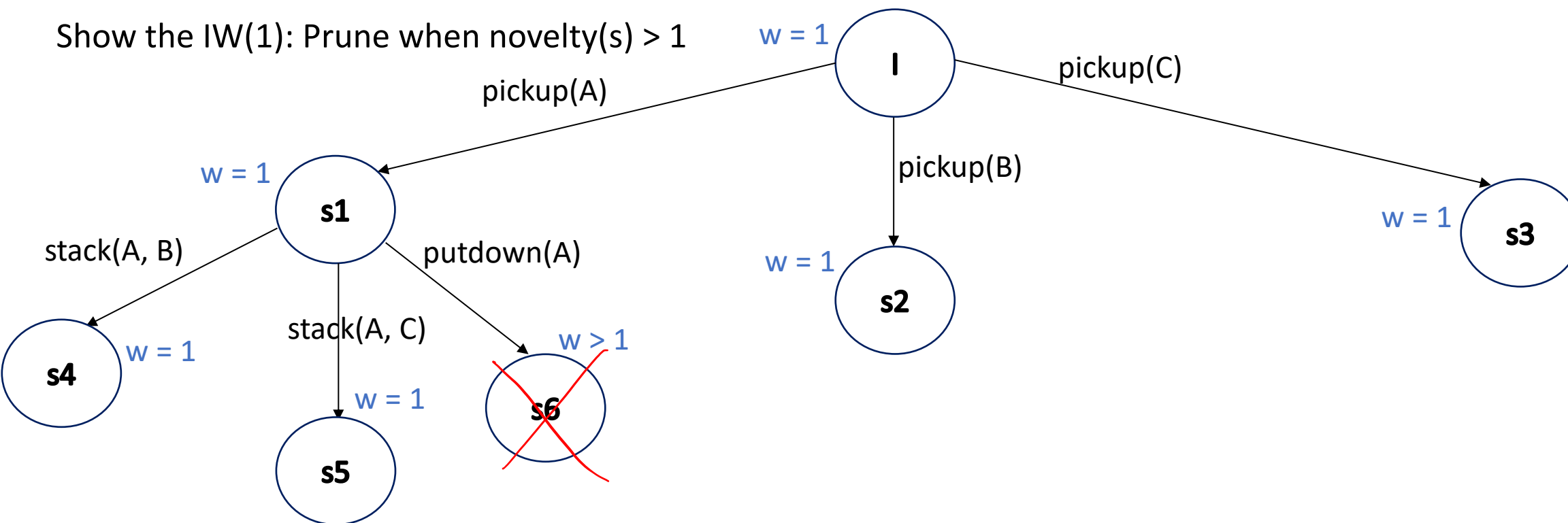
Novelty table

onTable(A), onTable(B), onTable(C), clear(A), clear(B), clear(C), handFree,
holding(A), holding(B), holding(C),
on(A, B), on(A, C)

Problem 2: Iterated Width (IW)

$I = \{\text{onTable(A)}, \text{onTable(B)}, \text{onTable(C)}, \text{clear(A)}, \text{clear(B)}, \text{clear(C)}, \text{handFree}\}$
 $G = \{\text{on(A, B)}\}$

Show the IW(1): Prune when novelty(s) > 1



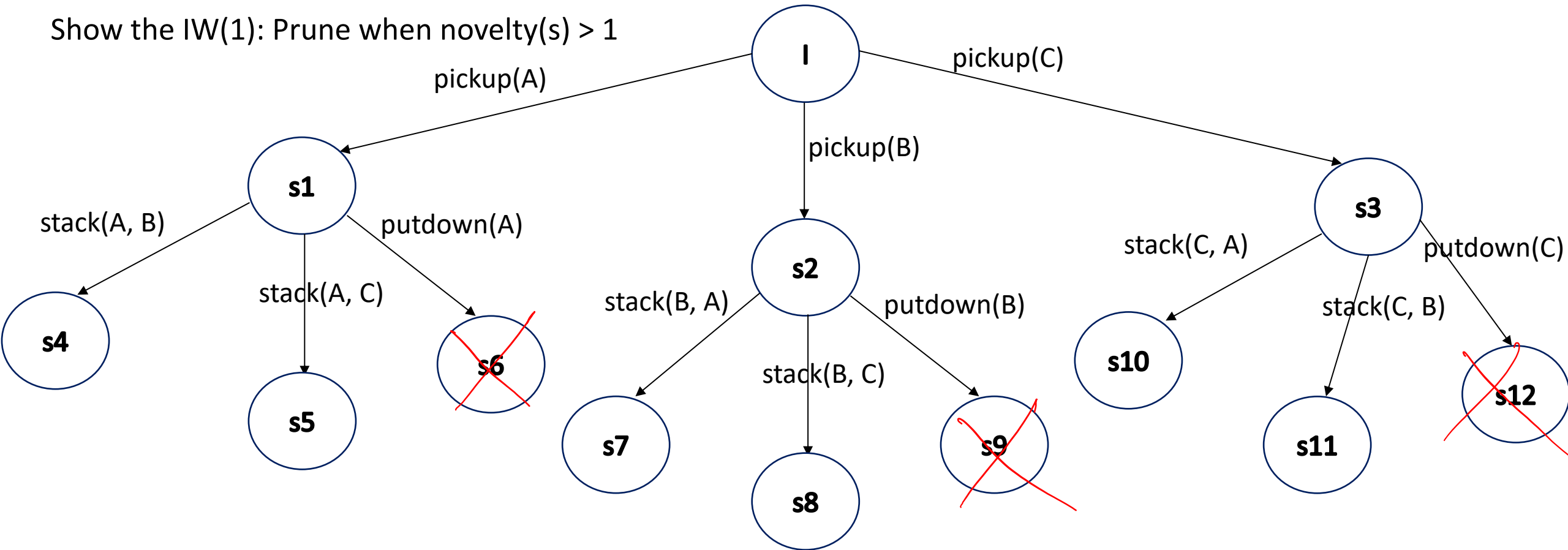
Novelty table

onTable(A), onTable(B), onTable(C), clear(A), clear(B), clear(C), handFree,
holding(A), holding(B), holding(C),
on(A, B), on(A, C)

Problem 2: Iterated Width (IW)

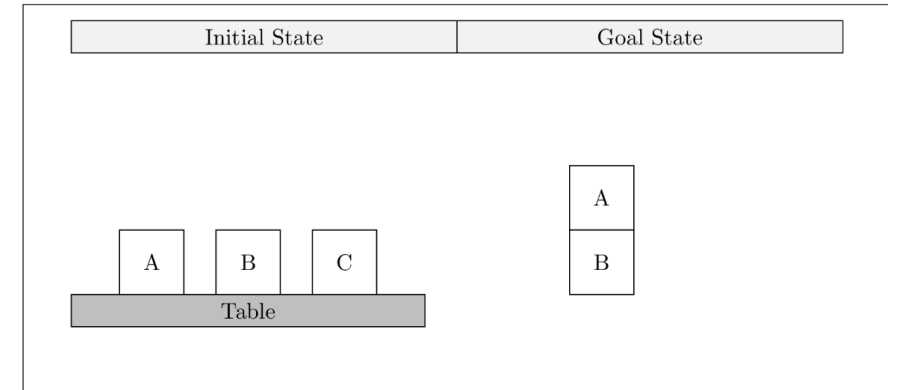
$I = \{\text{onTable(A)}, \text{onTable(B)}, \text{onTable(C)}, \text{clear(A)}, \text{clear(B)}, \text{clear(C)}, \text{handFree}\}$
 $G = \{\text{on(A, B)}\}$

Show the IW(1): Prune when novelty(s) > 1



Problem 2: Iterated Width (IW)

Task 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?



Problem 2: Iterated Width (IW)

Task 2: Can you think of an initial situation where IW(1) cannot find a solution for the goal $\text{on}(A,B)$, but IW(2) does, explain your answer?

