

The “Monotonicity” Relaxation

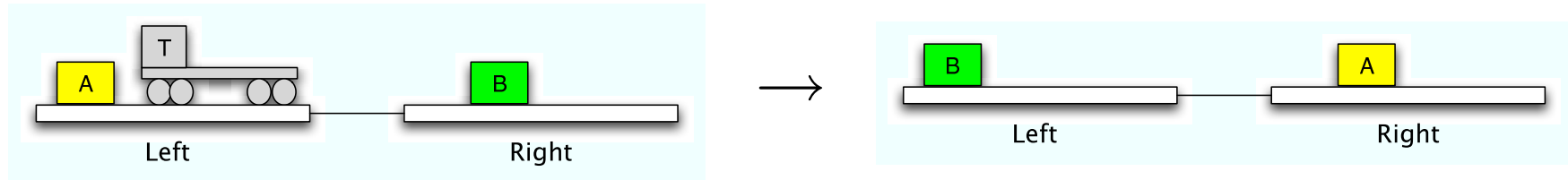
Ignoring Delete Lists

Let $\langle P, A, I, G \rangle$ be a Strips planning problem. For an action $a \in A$, the **monotonic relaxation** of a is $a^+ = (pre(a), add(a), \emptyset)$. For $A^+ = \{a^+ | a \in A\}$, $\langle P, A^+, I, G \rangle$ is the **monotonic relaxation** of $\langle P, A, I, G \rangle$.

Why monotonic?

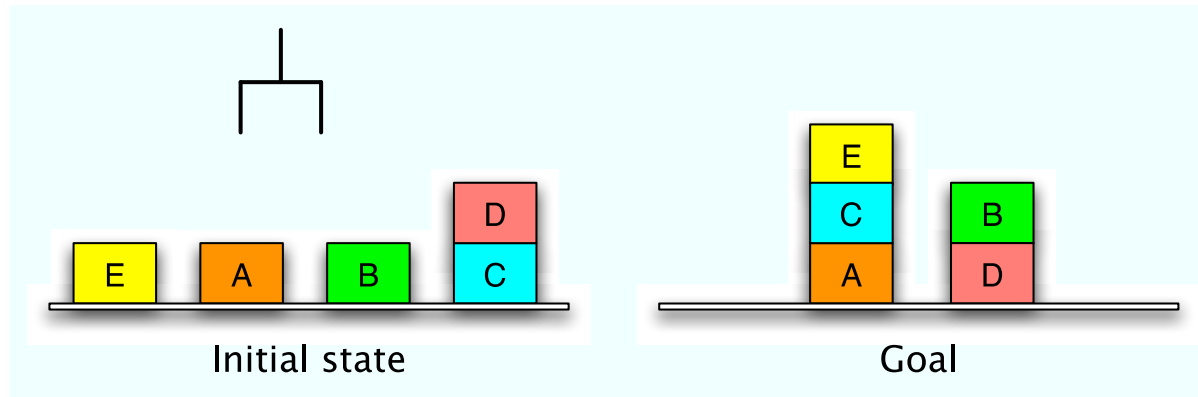
- All conditions in Strips are conjunctions of positive facts
- Under relaxed actions, the set of holding facts grows monotonically.
- “Achieved once stay forever”

Example: Logistics



- Initial state I : $\{at(A, Left), at(T, Left), at(B, Right)\}$
- $f(I, Drive(Left, Right)) = \{at(A, Left), at(T, Right), at(B, Right)\}$
- $f(I, Drive(Left, Right)^+) = \{at(A, Left), at(T, Left), at(T, Right), at(B, Right)\}$
- $f(I, \langle Drive(Left, Right), Load(A, Left) \rangle)$ is undefined
- $f(I, \langle Drive(Left, Right)^+, Load(A, Left)^+ \rangle) = \{at(A, Left), at(T, Left), at(T, Right), at(B, Right), in(A, T)\}$

Example: Blocksworld (with crane)



- Initial state I :

$\{on(E, Table), clear(E), \dots, on(C, Table), on(D, C), clear(D), holding(NIL)\}$

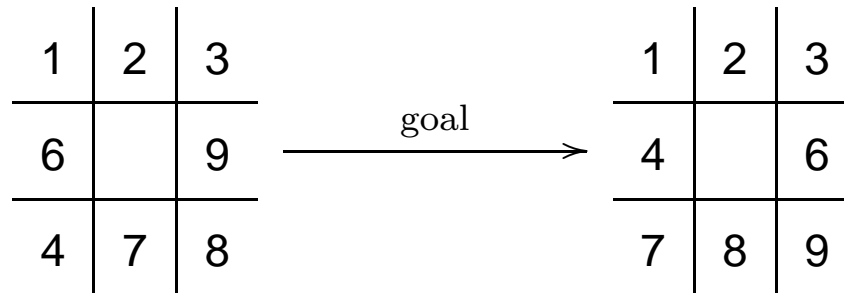
- $f(I, unstack(D, C)^+) =$

$\{on(E, Table), clear(E), \dots, on(C, Table), clear(C), holding(D), ???\}$

- $f(I, \langle unstack(D, C)^+, pickup(E)^+ \rangle) =$

$\{holding(E), ???, \dots, on(C, Table), clear(C), holding(D), ???\}$

Example: 8-Puzzle



- Real problem:
 - A tile can move from square A to square B if A is adjacent to B and B is blank
- Monotonically relaxed problem:
 - A tile can move from square A to square B if A is adjacent to B and B is blank
(!!!)
 - **In effect ...**

PLANSAT⁺

PLANSAT⁺: Given $\langle P, A, I, G \rangle$, is $\langle P, A^+, I, G \rangle$ solvable?

Proposition (Bylander, '94)

PLANSAT⁺ is in **P**

Proof:

$F \leftarrow I$

while $G \not\subseteq F$ **do**

$F' \leftarrow F \cup \{p \in P \mid \exists a \in A : pre(a) \subseteq F, p \in add(a)\}$

if $F' = F$ **then** *unsolvable*

$F \leftarrow F'$

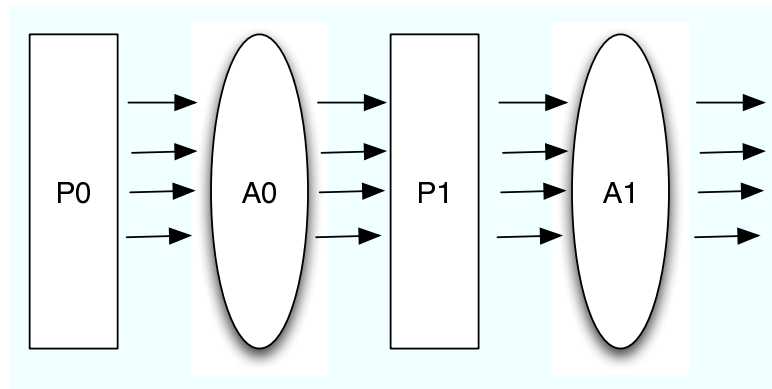
Graphical “interpretation” - Planning graphs

- Build **reachability graph** $P_0, A_0, P_1, A_1, \dots$

$$P_0 = \{p \in \text{Init}\}$$

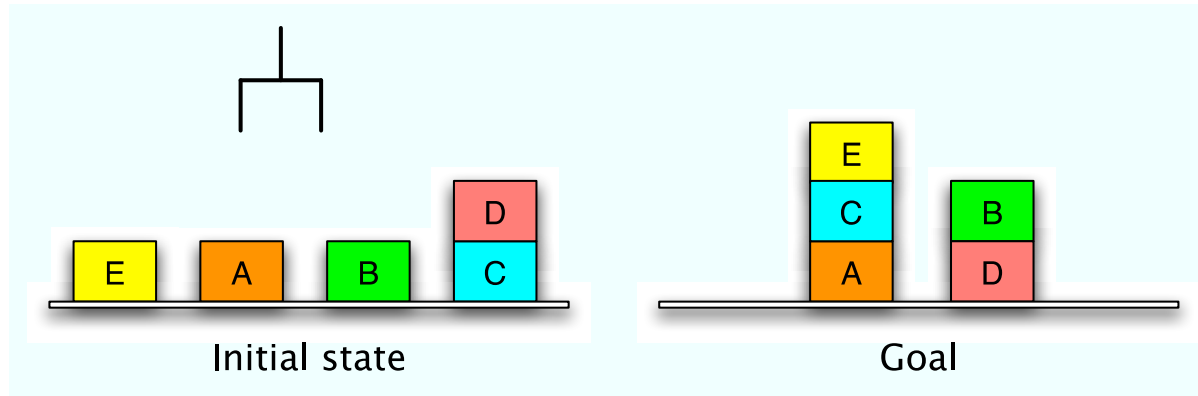
$$A_i = \{a \in O \mid \text{Prec}(a) \subseteq P_i\}$$

$$P_{i+1} = P_i \cup \{p \in \text{Add}(a) \mid a \in A_i\}$$



- Terminate when $G \subseteq P_i$

Example: Blocksworld



(1)

$\{on(E, Table), clear(E), on(A, Table), clear(A), on(B, Table), clear(B), on(C, Table), on(D, C), clear(D), holding(NIL)\}$

(1) $\{\dots, holding(E), holding(A), holding(B), holding(D), clear(C)\}$

(1) $\{\dots, holding(C), on(E, A), on(A, E), \dots\}$

(1) $\{\dots, on(C, A), \dots\}$

Blackboard: Planning graph for this example

Back to heuristics: h^+

Let $\langle P, A, I, G \rangle$ be a Strips problem. h^+ equals the length of an optimal plan for $\langle P, A^+, I, G \rangle$ or ∞ if $\langle P, A^+, I, G \rangle$ is unsolvable.

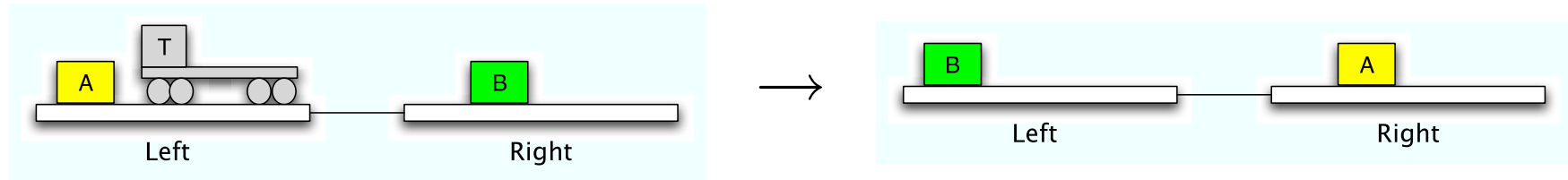
Standard terminology these days:

An (optimal) plan for $\langle P, A^+, I, G \rangle$ is called an **(optimal) relaxed plan** for $\langle P, A, I, G \rangle$

Proposition: Let $\langle P, A, I, G \rangle$ be a Strips problem. If $\langle a_1, \dots, a_k \rangle$ is a plan for $\langle P, A, I, G \rangle$, then $\langle a_1^+, \dots, a_k^+ \rangle$ is a plan for $\langle P, A^+, I, G \rangle$.

Corollary: For any $\langle P, A, I, G \rangle$, we have $h^+ \leq h^*$, that is h^+ is **admissible**

Example: Logistics



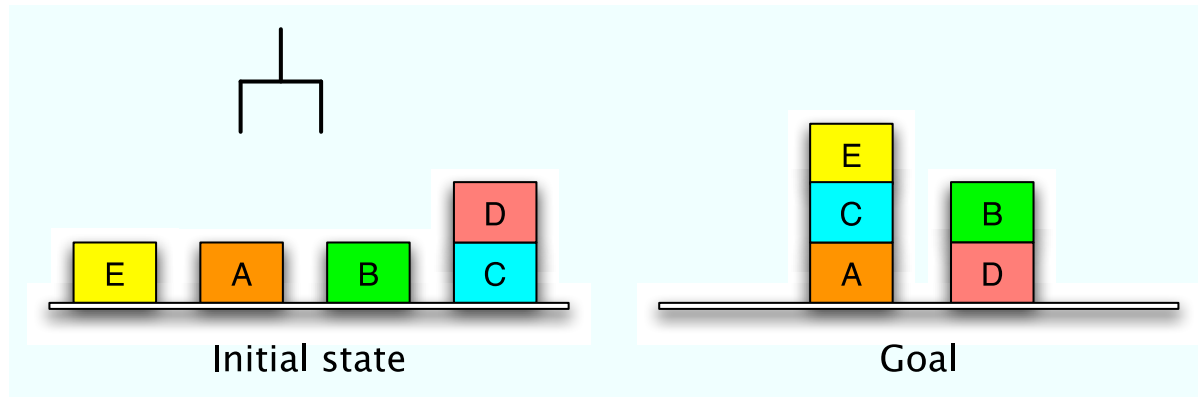
- Optimal plan:

1. $load(A, T, Left)$,
2. $drive(Left, Right)$,
3. $unload(A, T, Right)$,
4. $load(B, T, Right)$,
5. $drive(Right, Left)$,
6. $unload(B, T, Left)$

- Optimal relaxed plan: ??? (subsequence of the optimal plan)

- $h^*(I) = 6, h^+(I) = ???$

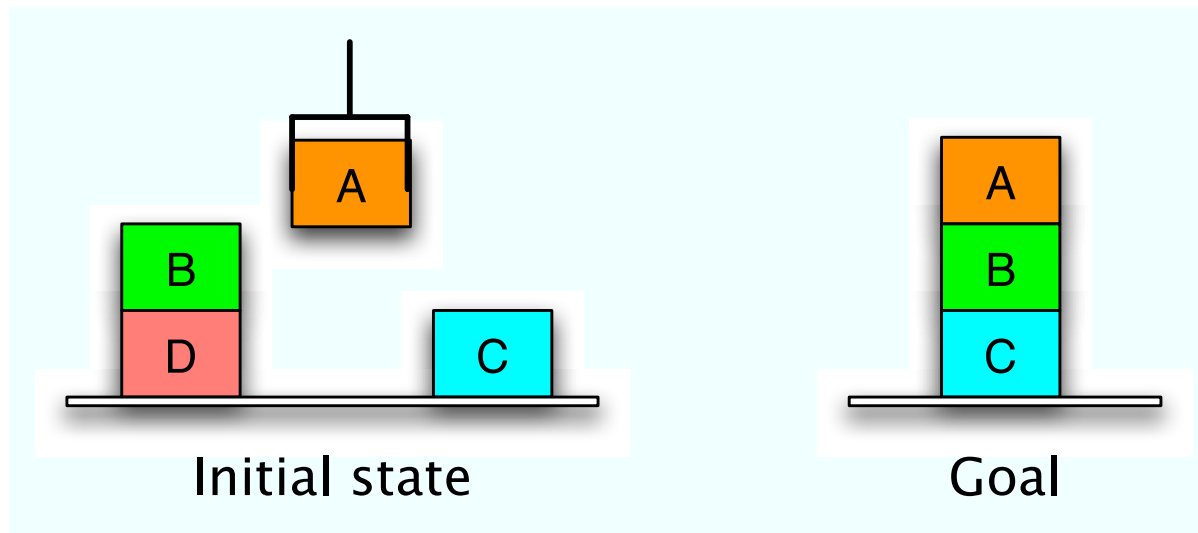
Example: Blocksworld



- Optimal plan: $\langle unstack(D, C), putdown(D), pickup(B), stack(B, D), pickup(C), stack(C, A), pickup(E), stack(E, C) \rangle$
- Optimal relaxed plan: ??? (subsequence of the optimal plan)
- $h^*(I) = 8, h^+(I) = ???$

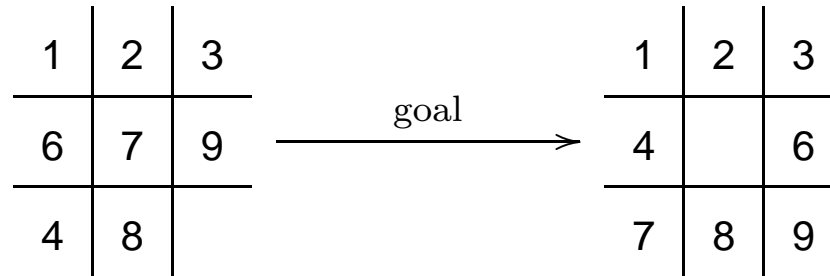
Always subsequence? (Just curious)

An optimal relaxed plan can *not* always be obtained by skipping actions from the (real) optimal plan.



- Optimal plan:
 $\langle \text{putdown}(A), \text{unstack}(B, D), \text{stack}(B, C), \text{pickup}(A), \text{stack}(A, B) \rangle$
- Optimal relaxed subsequence: ???
- Optimal relaxed plan: ???

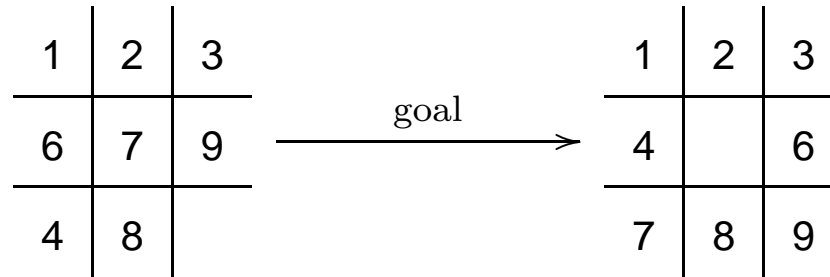
Example: 8-Puzzle



- A tile can move from square A to square B if A is adjacent to B and B is blank - solution distance h^*
- A tile can move from square A to square B if A is adjacent to B - manhattan distance heuristic h^{MD}
- A tile can move from square A to square B if A is adjacent to B and B is blank; in effect, the tile is at both A and B, and both A and B are blank - h^+

Here: $h^*(s_0) = 8$, $h^{MD}(s_0) = 6$, $h^{MT}(s_0) = ???$

Example: 8-Puzzle



Optimal MD plan:

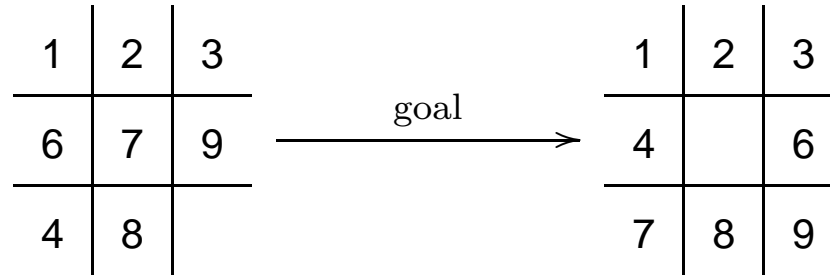
1. $move(t_9, p_6, p_9)$
2. $move(t_7, p_5, p_8)$
3. $move(t_6, p_4, p_5)$
4. $move(t_6, p_5, p_6)$
5. $move(t_4, p_7, p_4)$
6. $move(t_7, p_8, p_7)$

Optimal relaxed plan:

1. $move(t_9, p_6, p_9)$
2. $move(t_8, p_8, p_9)$
3. $move(t_7, p_5, p_8)$
4. $move(t_6, p_4, p_5)$
5. $move(t_6, p_5, p_6)$
6. $move(t_4, p_7, p_4)$
7. $move(t_7, p_8, p_7)$

So $h^*(s_0) = 8$, $h^{MD}(s_0) = 6$, $h^{MT}(s_0) = 7 (> h^{MD}!)$

8-Puzzle; h^+ vs. h^{MD}



h^+ **dominates** h^{MD}

- The goal is given as a conjunction of $at(t_i, p_j)$ atoms
- Achieving each single one of them takes at least as many steps as the respective tile's Manhattan distance
- Each action moves a single tile only

And we have just seen that h^+ **strictly dominates** h^{MD}

h^+ is cool! Oppss ...

Bounded-PLANSAT⁺: Given $\langle P, A, I, G \rangle$ and an integer b , is there a relaxed plan with $\leq b$ actions?

By computing optimal relaxed plans (for h^+) we would solve Bounded-PLANSAT⁺

Proposition (Bylander, '94)

Bounded-PLANSAT⁺ is in **NP**-complete

Proof: For instance, reduction from *Set-Cover* (Blackboard)

Approximating h^+ , Idea Nr. 1

Assume that there are no positive interactions, that is, all facts have to be achieved separately (Bonet et al., '97)

$$h_s^{add}(F) = \begin{cases} 0, & F \subseteq s \\ 1 + \min_{a, p \in add(a)} h_s^{add}(pre(a)), & F = \{p\} \\ \sum_{p \in s} h_s^{add}(p), & |F| > 1 \end{cases}$$

- For $\langle P, A, I, G \rangle$, we have $h^{add} = h_I^{add}(G)$
- This is a **pessimistic** simplification; it simplifies computation of heuristic, not solving the problem

Computing h^{add}

$h^{add}(p) \leftarrow 0$ for $p \in I$, $h^{add}(p) \leftarrow \infty$ for $p \notin I$

$changes \leftarrow \text{TRUE}$

while $changes$ **do**

$changes \leftarrow \text{FALSE}$

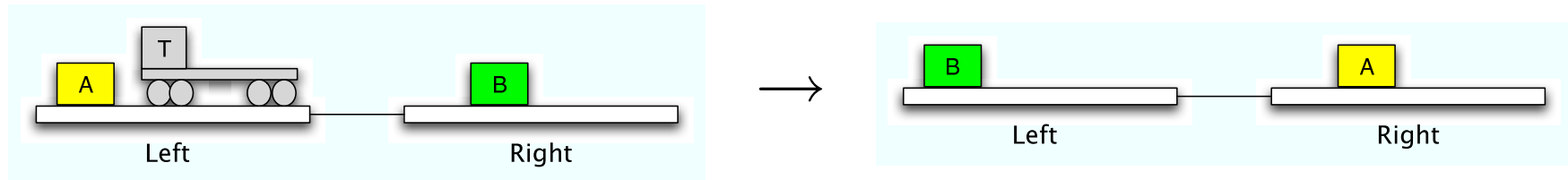
forall $a \in A$, $p \in add(a)$ **do**

if $1 + \sum_{p' \in pre(a)} h^{add}(\{p'\}) < h^{add}(\{p\})$ **then**

$h^{add}(\{p\}) \leftarrow 1 + \sum_{p' \in pre(a)} h^{add}(\{p'\})$

$changes \leftarrow \text{TRUE}$

Example: Logistics



- $h^{add} = 6 > h^+ = 5$
- $h^{add} = h^*$, but for the wrong reasons: $drive(Left, Right)$ is counted twice
- If we had 1000 packages, we would count it 1000 times

Every heuristic can be fooled, but

- It is *intuitive* that h^{add} overestimates too much in many (most?) natural examples, and
- Later we'll see that we can always do better

Approximating h^+ , Idea Nr. 2

Use the information collected by the “solvability” algorithm.

$F \leftarrow I$

while $G \not\subseteq F$ **do**

$F' \leftarrow F \cup \{p \in P \mid \exists a \in A : pre(a) \subseteq F, p \in add(a)\}$

if $F' = F$ **then** report *unsolvable*

$F \leftarrow F'$

- report *solvable*

Approximating h^+ , Exploring Idea Nr. 2

Use the information collected by the “solvability” algorithm.

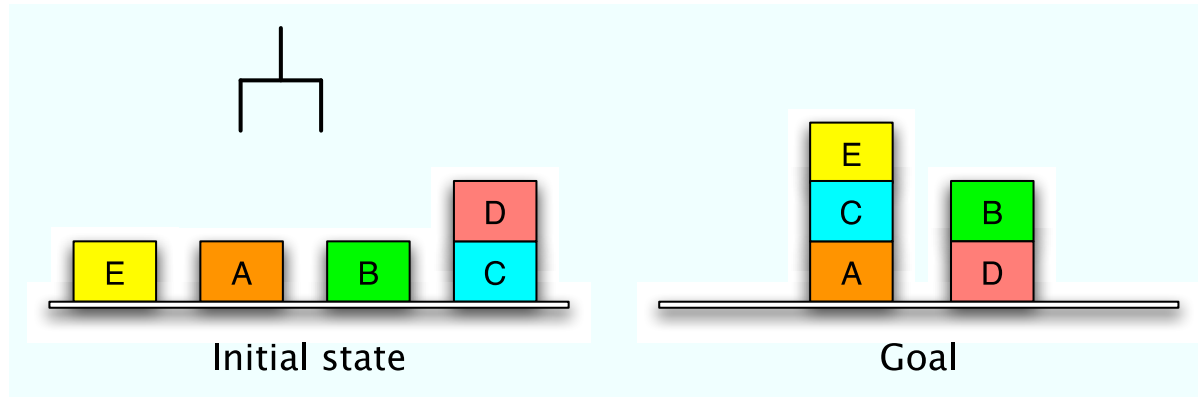
Approximate h^+ by the **number of iterations** until we get $G \subseteq F$

- Admissible (*right?*)
- Far too optimistic: it applies lots of actions in parallel
- *We'll come back to it later ...*

Approximate h^+ by the **number of actions used** until we get $G \subseteq F$

- Far too pessimistic

Approximating h^+ , Exploring Idea Nr. 2



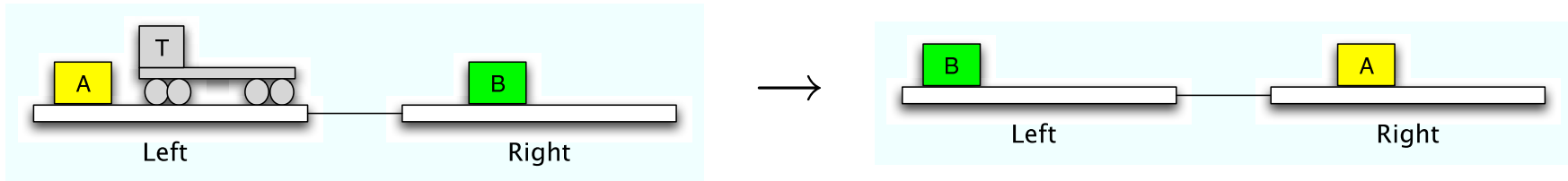
- $h^* = 8, h^+ = 7$
- $estimate_{2a} = 3$
- $estimate_{2b} = 46$

Approximating h^+ , Idea Nr. 3

Find some, not necessarily optimal, relaxed plan for $\langle P, A, I, G \rangle$ and take the length of that relaxed plan as (over)estimate of h^+ (Hoffman & Nebel, '01)

- This estimated is usually named h^{FF} after the first planner that used it.
- Two-step process to compute h^{FF}
 - Chain forward to build a **relaxed planning graph** (RPG)
 - Chain backward from G to extract a relaxed plan from the RPG

Example: Logistics



Blackboard

Fooling h^{FF}

h^{FF} is not an admissible estimate

- Action selection in backchaining

- $P = \{p_1, p_2\}, I = \emptyset, G = \{p_1, p_2\}$
- $A = \{a_{all} = (\emptyset, \{p_1, p_2\}, \emptyset), a_1 = (\emptyset, \{p_1\}, \emptyset), a_2 = (\emptyset, p_2, \emptyset)\}$
- All actions appear in the first action layer of RPG, but the plan extraction procedure may choose $\{a_1, a_2\}$

- Optimal plan may not be in RPG

- $P = \{p_1, p_2, p_3\}, I = \emptyset, G = \{p_1, p_2, p_3\}$
- $A = \{a_1 = (\emptyset, \{p_1\}, \emptyset), a_2 = (\emptyset, p_2, \emptyset), a_3 = (\emptyset, \{p_3\}, \emptyset), a_4 = (\{p_1\}, \{p_2, p_3\}, \emptyset)\}$
- a_4 is not in RPG (*right?*), but the optimal plan is $\langle a_1, a_4 \rangle$

Overview

h^+ Length of an optimal relaxed plan; admissible; NP-hard to compute

h^{add} Assume that all facts must be achieved completely independently; not admissible, over-estimates often; compute by forward chaining over values for single facts

h^{FF} Approximate h^+ by length of some relaxed plan; not admissible, over-estimates rarely; compute by forward chaining (RPG) and backward chaining (relaxed plan extraction)

Non-optimal planning these days:

- use h^{FF} to find (hopefully) short a relaxed plan without optimality guarantee
- most currently successful non-optimal systems do variations of this
- search procedure: something light yet sophisticated enough: variants of hill-climbing

Literature

- T. Bylander, The Computational Complexity of Propositional STRIPS Planning, Artificial Intelligence Journal, 1994
- B. Bonet, G. Loerincs, and H. Geffner, A Robust and Fast Action Selection Mechanism for Planning, AAAI-97
- Jörg Hoffmann and Bernhard Nebel, The FF Planning System: Fast Plan Generation Through Heuristic Search, Journal of Artificial Intelligence Research, 2001
- B. Bonet and H. Geffner, Planning as Heuristic Search, Artificial Intelligence Journal, 2001 (also for the next lecture)