



Plan Generation & Causal-Link Planning

José Luis Ambite



Review

- Frame problem:
 - How to specify what does not change
- Qualification problem
 - Hard to specify all preconditions
- Ramification problem
 - Hard to specify all effects



Review

- Situation Calculus

- Successor state axioms:

$$\text{Broken}(x, \text{do}(s,a)) \leftrightarrow [a = \text{drop}(x) \wedge \text{fragile}(x, s)] \vee \\ \exists b[a = \text{explode}(b) \wedge \text{nextTo}(b, x, s)] \vee \\ \text{broken}(x,s) \wedge \neg \exists a = \text{repair}(x)$$

- Preconditions axioms:

$$\text{Poss}(\text{pickup}(r,x), s) \leftrightarrow \text{robot}(r) \wedge \forall z \neg \text{holding}(z, x, s) \wedge \\ \text{nextTo}(r, x, s)$$

- Strips representation

- Means-ends analysis

- Networks of Actions (Noah)



Domain-Independent Planning

Inputs:

- Domain Action Theory
- Problem Instance
 - Description of (initial) state of the world
 - Specification of desired goal behavior

Output: sequence of actions that
executed in initial state satisfy goal

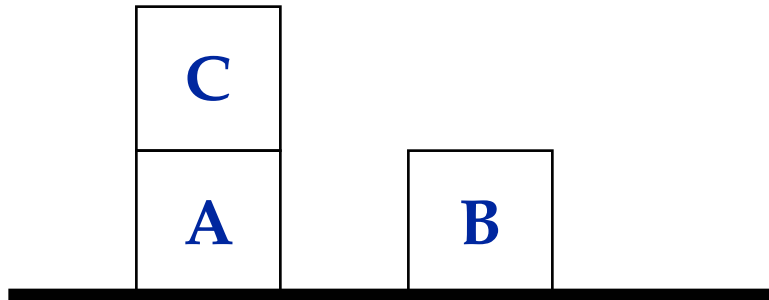


“Classical Planning” Assumptions

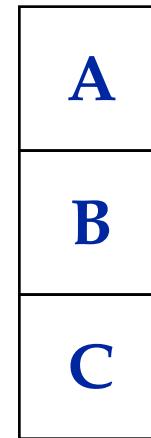
- Atomic Time
 - Instantaneous actions
- Deterministic Effects
- Omniscience
- Sole agent of change
- Goals of attainment

Example Problem Instance: "Sussman Anomaly"

Initial State:



Goal:

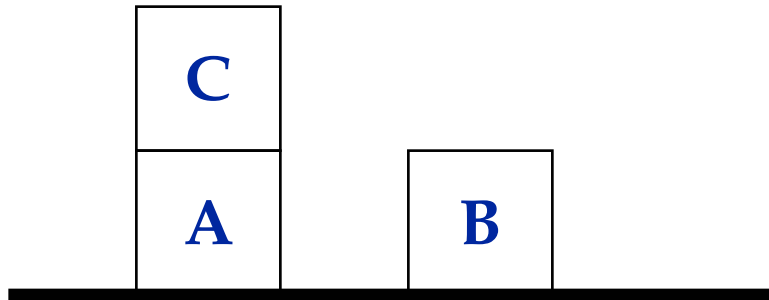


Initial State: (on-table A) (on C A) (on-table B)
(clear B) (clear C)

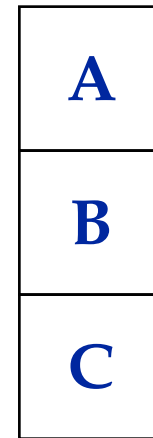
Goal: (on A B) (on B C)

Example Problem Instance: "Sussman Anomaly"

Initial State:



Goal:



Initial State: (and (on-table A) (on C A)
(on-table B) (clear B) (clear C))

Goal: (and (on A B) (on B C))



Action Representation: Propositional STRIPS

Move-C-from-A-to-Table:

preconditions: (on C A) (clear C)

effects:

add (on-table C)

delete (on C A)

add (clear A)

Solution to frame problem: explicit effects are the only changes to the state.



Action Representation: Propositional STRIPS

Move-C-from-A-to-Table:

preconditions: (and (on C A) (clear C))

effects:

(and (on-table C)

(not (on C A))

(clear A))

Solution to frame problem: explicit effects are the only changes to the state.

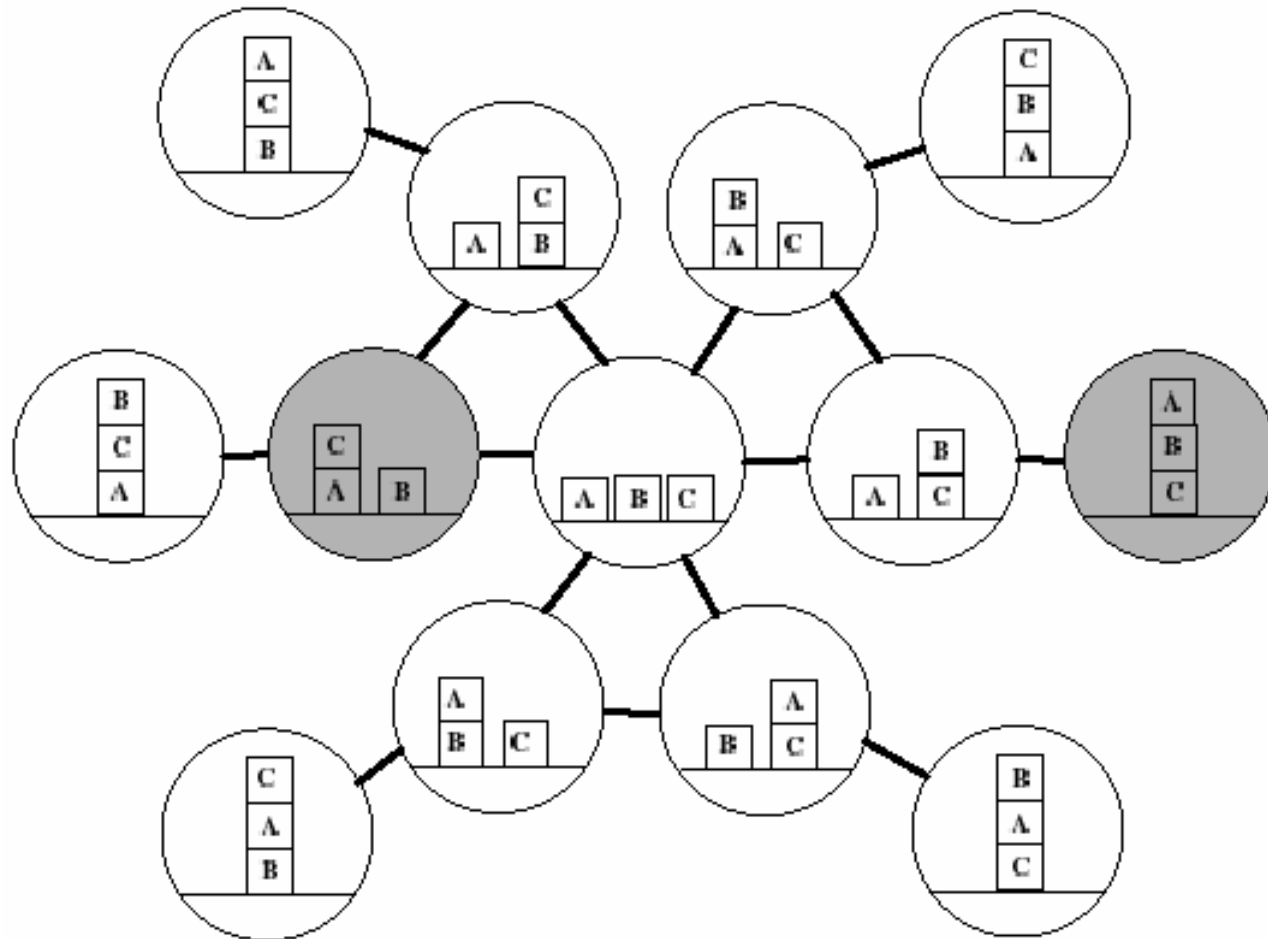


Plan Generation: Search space of world states

Planning as a (graph) search problem

- Nodes: world states
- Arcs: actions
- Solution: path from the initial state to one state that satisfies the goal
 - Initial state is fully specified
 - There are many goal states

Search Space: Blocks World



Progression:

Forward search (I \rightarrow G)

ProgWS(state, goals, actions, path)

If state satisfies goals, then return path

else a = **choose**(actions), s.t.

 preconditions(a) satisfied in state

if no such a, then return failure

else return

ProgWS(apply(a, state), goals, actions,
 concatenate(path, a))

First call: ProgWS(IS, G, Actions, ())

Progression Example: Sussman Anomaly

I: (on-table A) (on C A) (on-table B) (clear B) (clear C)

G: (on A B) (on B C)

- P(I, G, BlocksWorldActions, ())
- P(S1, G, BWA, (move-C-from-A-to-table))
- P(S2, G, BWA, (move-C-from-A-to-table, move-B-from-table-to-C))
- P(S3, G, BWA, (move-C-from-A-to-table, move-B-from-table-to-C, move-A-from-table-to-B))

**Non-Deterministic
Choice!**

$G \subseteq S3 \Rightarrow$ Success!

Regression:

Backward Search ($I \leftarrow G$)

RegWS(init-state, current-goals, actions, path)

If init-state satisfies current-goals, then return path

else $a = \mathbf{choose}$ (actions), s.t. some effect of a satisfies one of current-goals

If no such a , then return failure [unachievable*]

If some effect of a contradicts some of current-goals, then return failure [inconsistent state]

$CG' = \text{current-goals} - \text{effects}(a) + \text{preconditions}(a)$

If $\text{current-goals} \subset CG'$, then return failure [useless*]

RegWS(init-state, CG' , actions, concatenate(a , path))

First call: RegWS(IS, G, Actions, ())

Regression Example: Sussman Anomaly

I: (on-table A) (on C A) (on-table B) (clear B) (clear C)

G: (on A B) (on B C)

- R(I, G, BlocksWorldActions, ())
- R(I, ((clear A) (on-table A) (clear B) (on B C)), BWA, (move-A-from-table-to-B))
- R(I, ((clear A) (on-table A) (clear B) (clear C), (on-table B)), BWA, (move-B-from-table-to-C, move-A-from-table-to-B))
- R(I, ((on-table A) (clear B) (clear C) (on-table B) (on C A)), BWA, (move-C-from-A-to-table, move-B-from-table-to-C, move-A-from-table-to-B))

**Non-Deterministic
Choice!**

current-goals \subseteq I \Rightarrow Success!

Regression Example: Sussman Anomaly

I: (on-table A) (on C A) (on-table B) (clear B) (clear C)

G: (on A B) (on B C)

- P(I, G, BlocksWorldActions, ())
- P(S1, G, BWA, (move-C-from-A-to-table))
- P(S2, G, BWA, (move-C-from-A-to-table, move-B-from-table-to-C))
- P(S3, G, BWA, (move-C-from-A-to-table, move-B-from-table-to-C, move-A-from-table-to-B))

**Non-Deterministic
Choice!**

$G \subseteq S3 \Rightarrow$ Success!

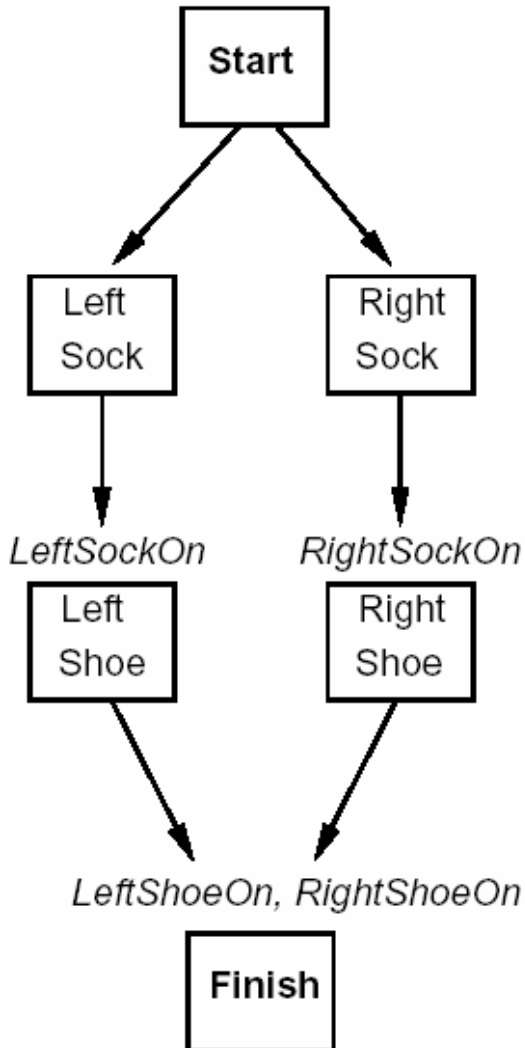


Progression vs. Regression

- Both algorithms are:
 - Sound: the result plan is valid
 - Complete: if valid plan exists, they find one
- Non-deterministic choice => search!
 - Brute force: DFS, BFS, Iterative Deepening, ...
 - Heuristic: A*, IDA*, ...
- Complexity: $O(b^n)$ worst-case
b = branching factor, n = |"choose"|
- Regression: often smaller b, focused by goals
- Progression: full state to compute heuristics

Total-Order vs Partial-Order Plans

Partial Order Plan:



Total Order Plans:





Plan Generation: Search space of plans

Partial-Order Planning (POP)

- Nodes are partial plans
- Arcs/Transitions are plan refinements
- Solution is a node (not a path).

Principle of “Least commitment”

- e.g. do not commit to an order of actions until it is required



Partial Plan Representation

- Plan = (A, O, L), where
 - A: set of actions in the plan
 - O: *temporal orderings* between actions ($a < b$)
 - L: *causal links* linking actions via a literal
- Causal Link: $A_p \xrightarrow{Q} A_c$

Action A_c (consumer) has precondition Q that is established in the plan by A_p (producer).

move-a-from-b-to-table $\xrightarrow{\text{(clear b)}}$ move-c-from-d-to-b



Threats to causal links

Step A_t threatens link (A_p, Q, A_c) if:

1. A_t has (not Q) as an effect, and
2. A_t could come between A_p and A_c , i.e.
 $O \cup (A_p < A_t < A_c)$ is consistent

What's an example of an action that threatens the link example from the last slide?



Initial Plan

For uniformity, represent initial state and goal with two special actions:

- A_0 :
 - no preconditions,
 - initial state as **effects**,
 - must be the **first** step in the plan.
- A_∞ :
 - no effects
 - goals as **preconditions**
 - must be the **last** step in the plan.



POP algorithm

POP((A, O, L), agenda, actions)

If agenda = () then return (A, O, L)

Pick (Q, a_{need}) from agenda

a_{add} = **choose**(actions) s.t. Q ∈ effects(a_{add})

If no such action a_{add} exists, **fail**.

L' := L ∪ (a_{add}, Q, a_{need}) ; O' := O ∪ (a_{add} < a_{need})

agenda' := agenda - (Q, a_{need})

If a_{add} is new, then A := A ∪ a_{add} and

∀P ∈ preconditions(a_{add}), add (P, a_{add}) to agenda'

For every action a_t that threatens any causal link (a_p, Q, a_c) in L'

choose to add a_t < a_p or a_c < a_t to O.

If neither choice is consistent, **fail**.

POP((A', O', L'), agenda, actions)

Termination

Goal Selection

Action Selection

Update goals

Protect causal links

- Demotion: a_t < a_p

- Promotion: a_c < a_t



POP

POP is sound and complete

- POP Plan is a solution if:
 - All preconditions are supported (by causal links), i.e., no open conditions.
 - No threats
 - Consistent temporal ordering
- By construction, the POP algorithm reaches a solution plan

POP example: Sussman Anomaly

(on C A) (on-table A) A0 (on-table B) (clear C) (clear B)

(on A B) (on B C)
Ainf

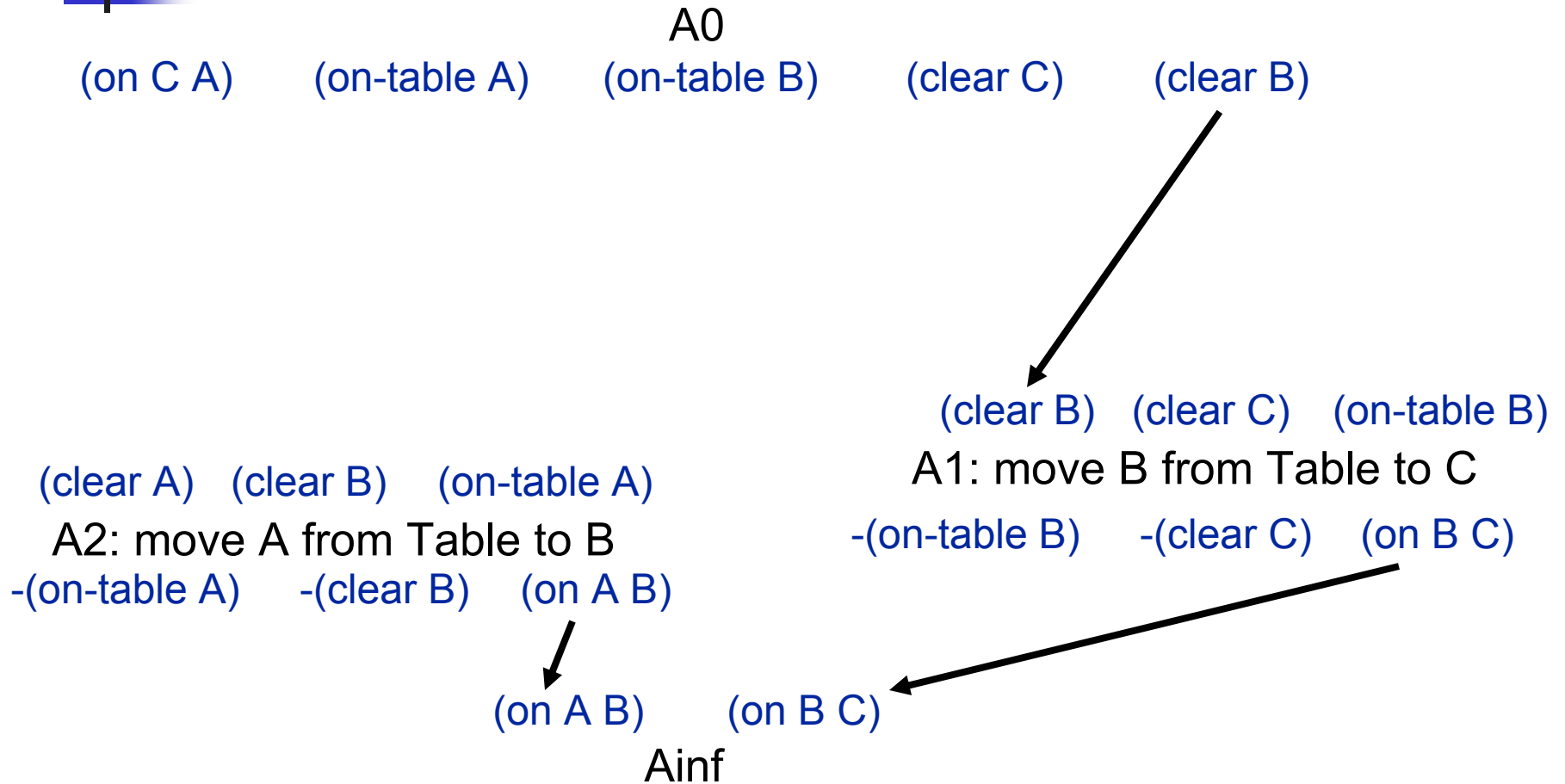
Work on open precondition (on B C) and (clear B)

A0
(on C A) (on-table A) (on-table B) (clear C) (clear B)

(clear B) (clear C) (on-table B)
A1: move B from Table to C
-(on-table B) -(clear C) (on B C)

(on A B) (on B C)
Ainf

Work on open precondition (on A B)



Protect causal links

A0

(on C A) (on-table A) (on-table B) (clear C) (clear B)

(A0, (clear B), A1) threatened by A2
--> Promotion

A1: move B from Table to C

-(on-table B) -(clear C) (on B C)

(clear A) (clear B) (on-table A)

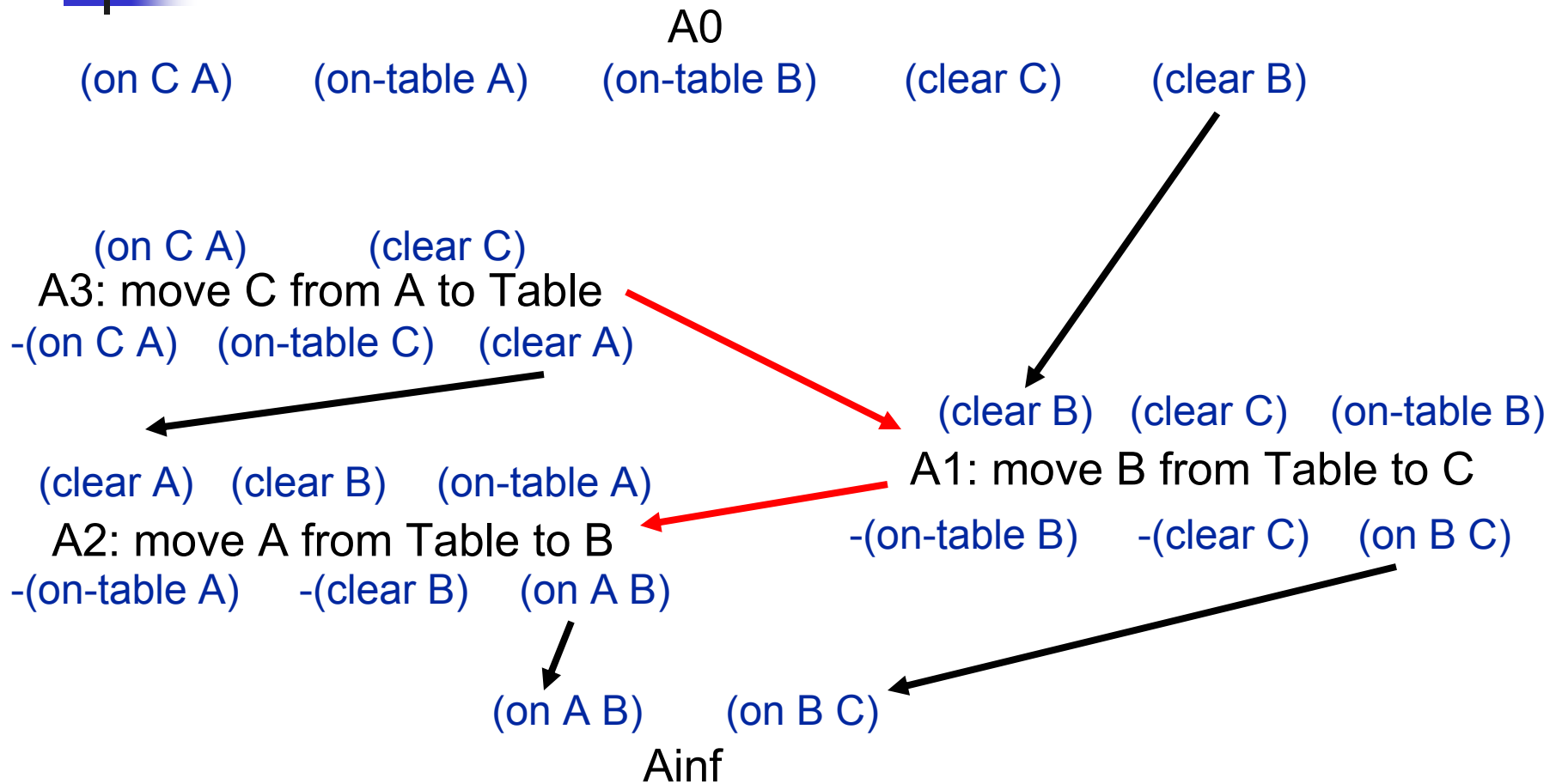
A2: move A from Table to B

-(on-table A) -(clear B) (on A B)

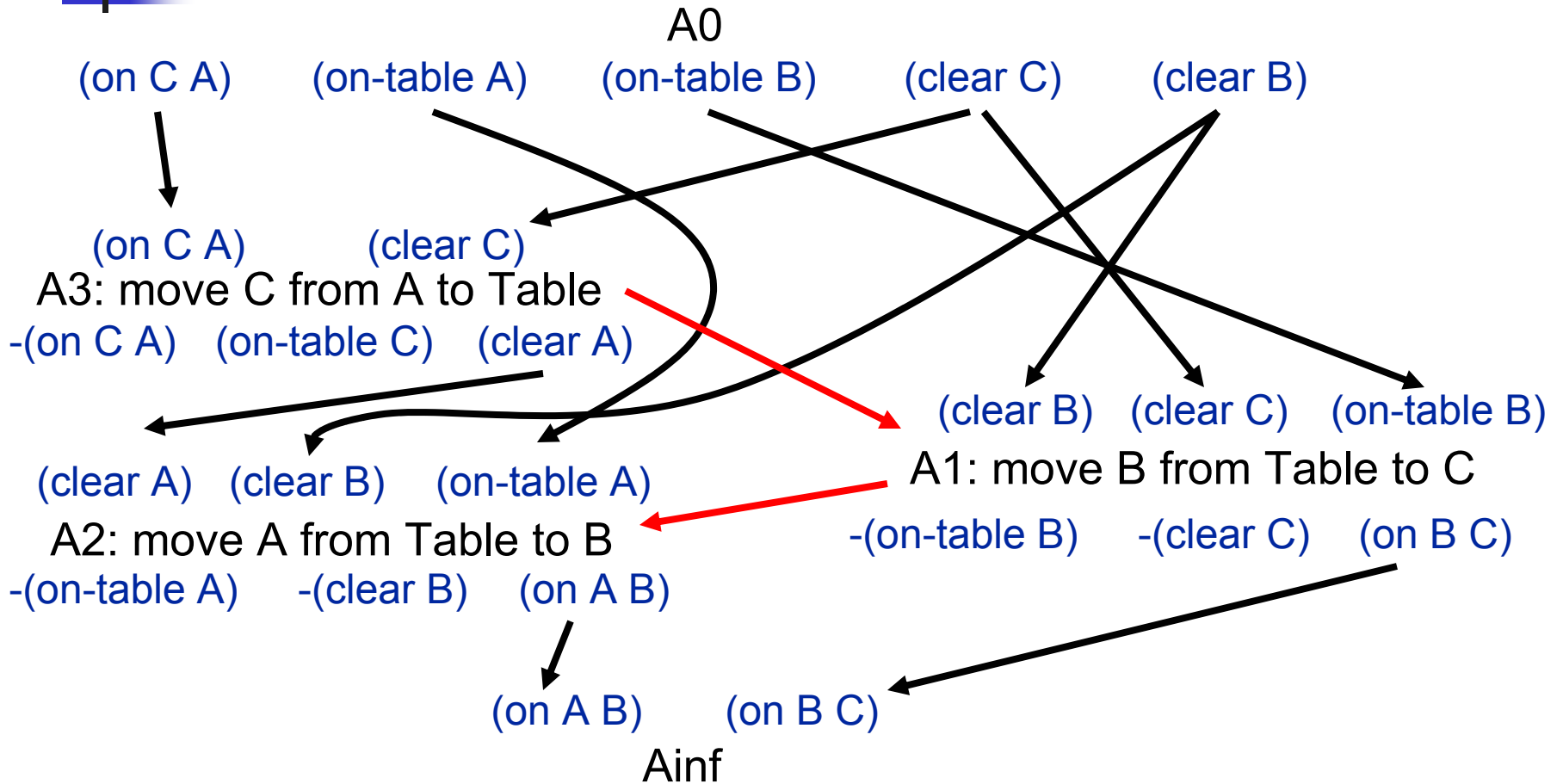
(on A B) (on B C)

Ainf

Work on open precondition (clear A) and protect links



Final plan





Partial-Order Planning vs State-Space Planning

Complexity: $O(b^n)$ worst-case

- Non-deterministic choices (n):
 - ProgWS, RegWS: $n = |\text{actions}|$
 - POP: $n = |\text{preconditions}| + |\text{link protection}|$
 - Generally an action has several preconditions
- Branching factor (b)
POP has smaller b :
 - No backtrack due to goal ordering
 - Least commitment: no premature step ordering
 - Does POP make the least possible amount of commitment?