Least Commitment Planning

Stephen G. Ware

CSCI 4525 / 5525

Narrative Intelligence LAB

THE UNIVERSITY of NEW ORLEANS

Motivation

Before the advent of accurate planning heuristics, most planning research focused on reducing the branching factor of search through abstraction.

The idea: allow 1 action in the search to represent as many actions in a plan as possible.

Cargo Domain

- The world consists of cargo, airplanes, and airports.
- The initial state specifies where each plane and each cargo is.
- There are three actions:
  - Load cargo onto a plane at the same airport.
  - Unload cargo from a plane to the current airport.
  - Fly a plane (and its cargo) to another airport.

Typed Constants

Rather than add unary predicates such as $block(B)$ to the initial state which never change, assign each constant and each variable a type.

Variables of a certain type can only bind to constants of that same type.

Cargo Domain Action Templates

Action: $load(Cargo\ c, Plane\ p, Airport\ a)$
Precondition: $at(c, a) \land at(p, a)$
Effect: $in(c, p) \land \neg at(c, a)$

Action: $unload(Cargo\ c, Plane\ p, Airport\ a)$
Precondition: $in(c, p) \land at(p, a)$
Effect: $at(c, a) \land \neg in(c, p)$

Action: $fly(Plane\ p, Airport\ from, Airport\ to)$
Precondition: $at(p, from)$
Effect: $at(p, to) \land \neg at(p, from)$

Cargo Problem

Constants:

- 2 Planes, P1 and P2.
- 2 Cargos, C1 and C2.
- 2 Airports, MSY and ATL.

Initial State: $at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

Goal: $at(C1, MSY)$

Cargo Problem

Initial State: $at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

Goal: $at(C1, MSY)$

How many actions are applicable in the initial state?

1. $load(C1, P1, ATL)$
2. $load(C2, P1, ATL)$
3. $load(C1, P2, ATL)$
4. $load(C2, P2, ATL)$
5. $fly(P1, ATL, MSY)$
6. $fly(P2, ATL, MSY)$
7. $fly(P1, ATL, ATL)$
8. $fly(P2, ATL, ATL)$

Cargo Problem

Initial State: $at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

Goal: $at(C1, MSY)$

How many actions can achieve the goal?

1. $unload(C1, P1, MSY)$
2. $unload(C1, P2, MSY)$

Backwards Search

Rather than searching forward from the initial state toward the goal, search backward from the goal to the initial state.

The Null Plan

Idea: Goals and preconditions are generally handled the same way. The same is true of the initial state and effects.

The **null plan** is a plan with 2 dummy steps. The **start step** has no preconditions and the initial state as its effects. The **end step** has the goal as its preconditions and no effect.

The Null Plan

$(no\ preconditions)$

$start$

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

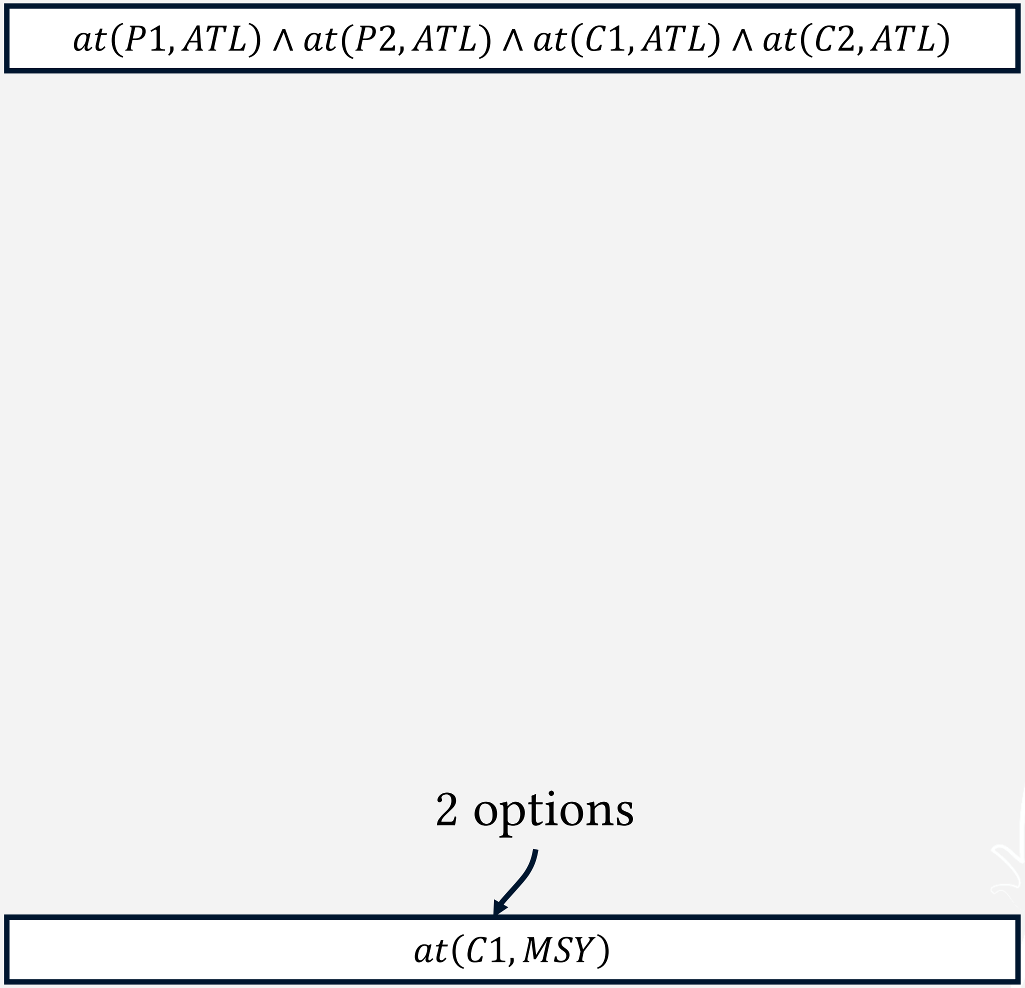
$at(C1, MSY)$

end

$(no\ effects)$

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

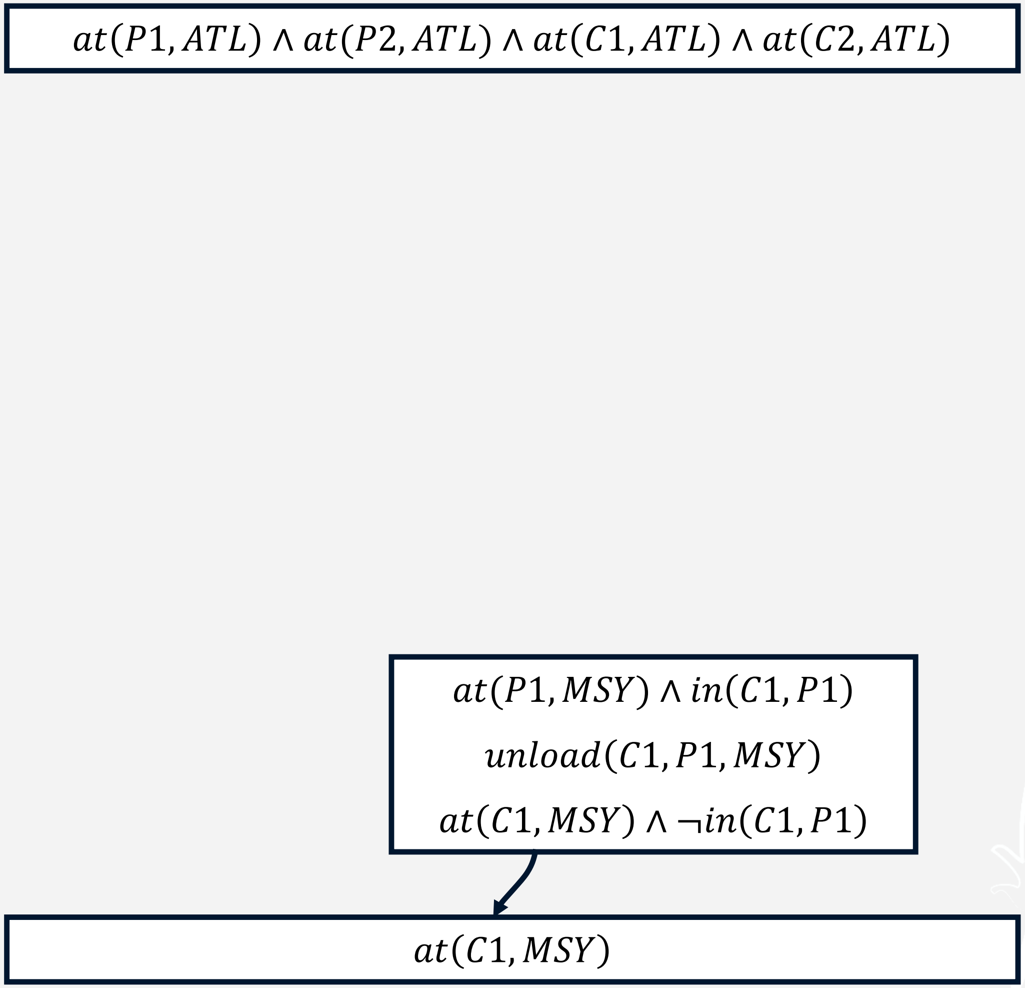
$at(C1, MSY)$



$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

2 options

$at(C1, MSY)$



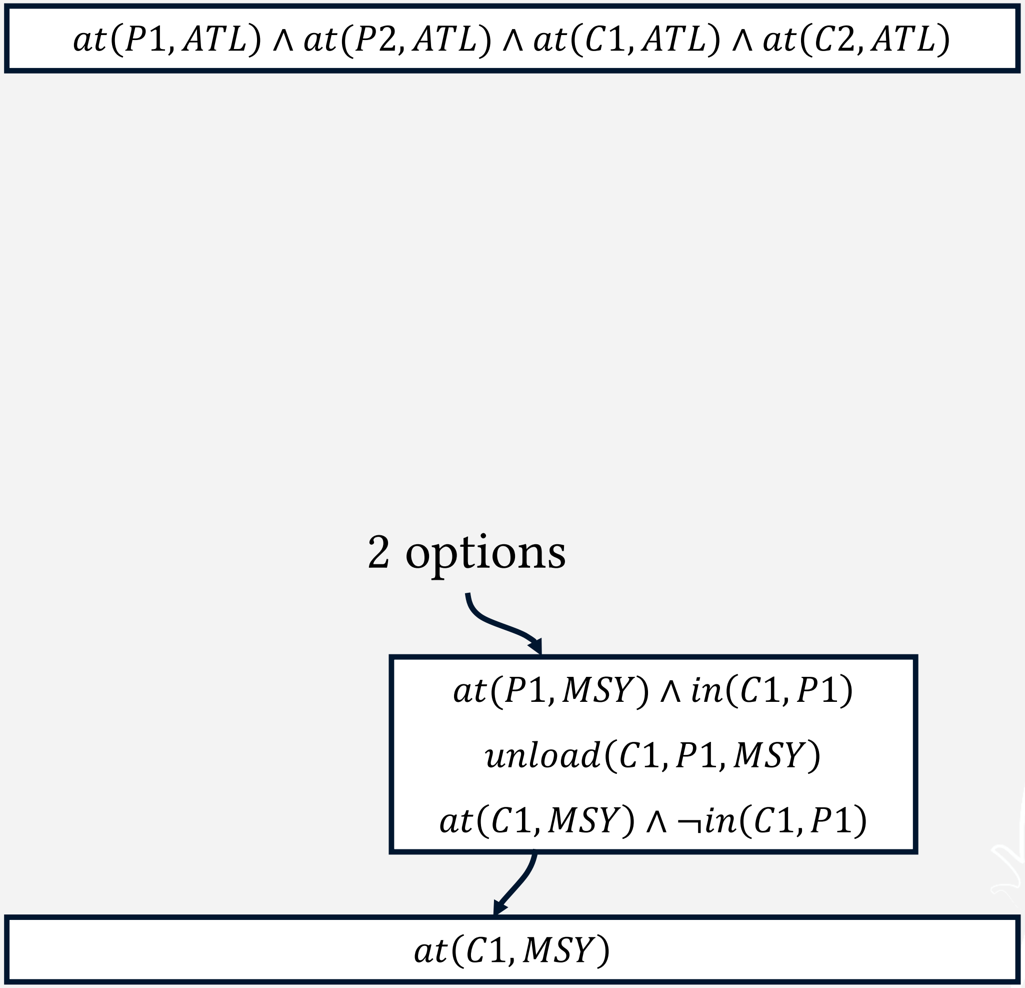
$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$



$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

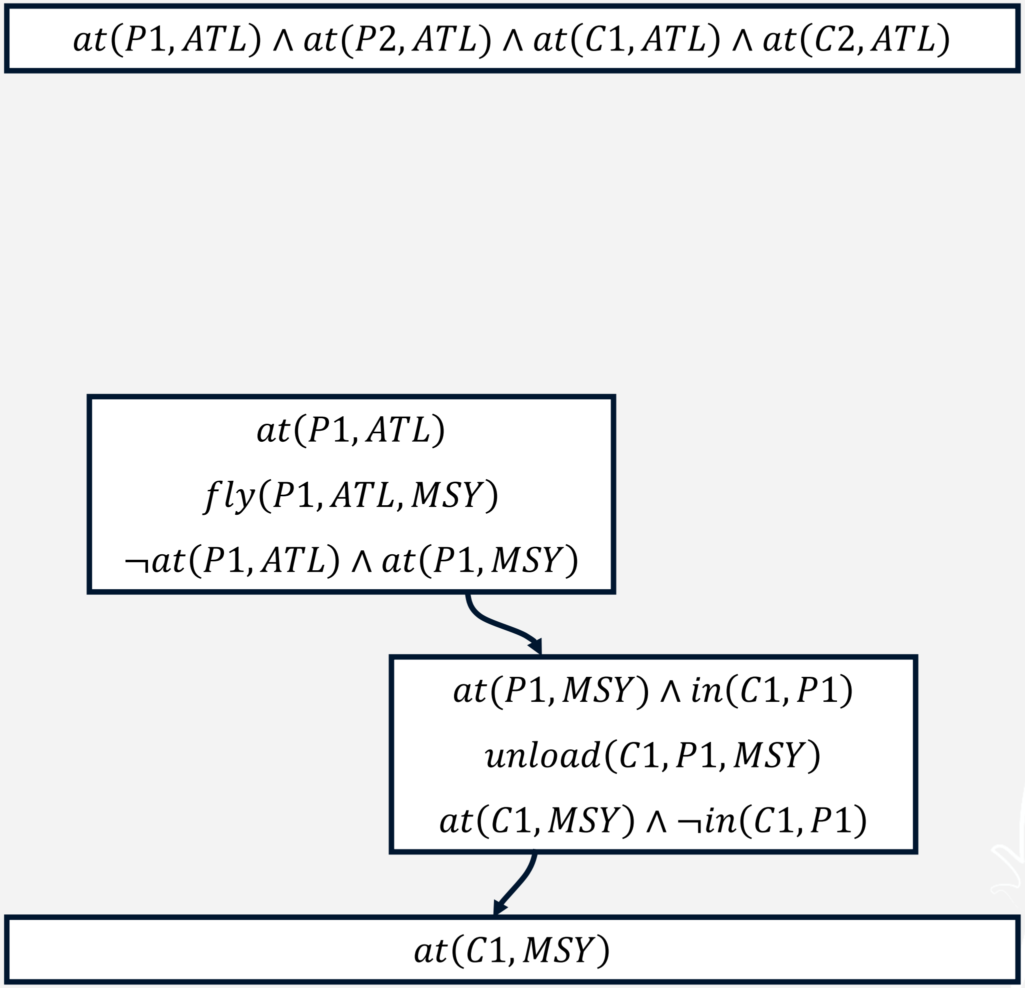
2 options

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$



$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

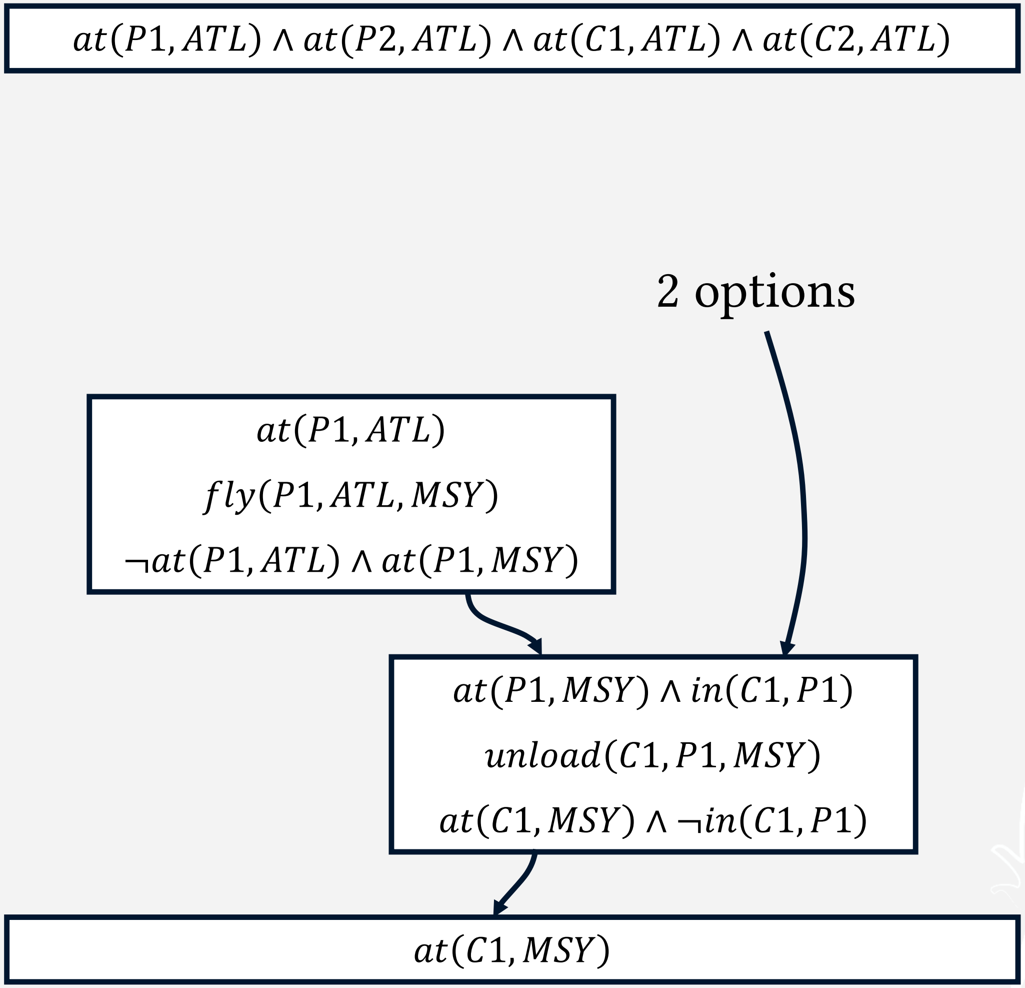
$\neg at(P1, ATL) \land at(P1, MSY)$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

$at(C1, MSY)$



$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

2 options

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

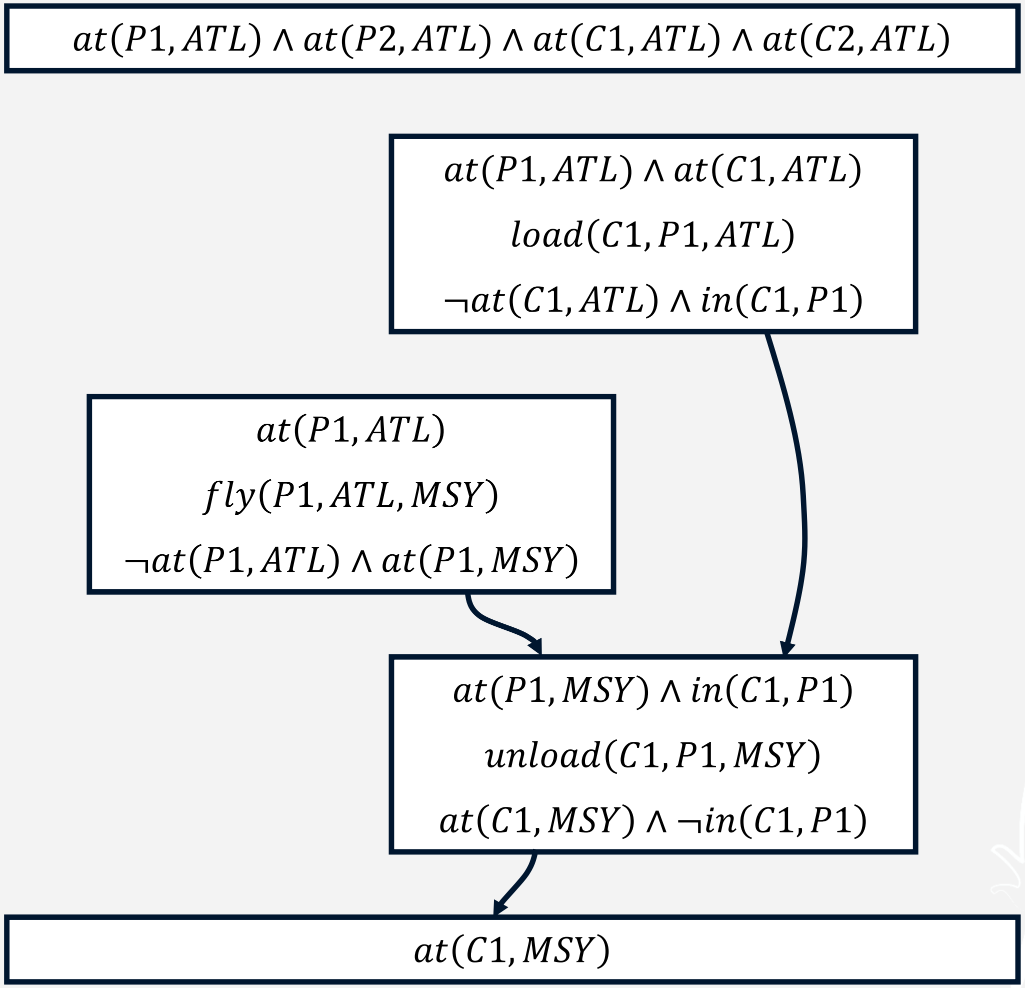
$\neg at(P1, ATL) \land at(P1, MSY)$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

$at(C1, MSY)$



$at(P1,ATL) \wedge at(P2,ATL) \wedge at(C1,ATL) \wedge at(C2,ATL)$

$at(P1,ATL) \wedge at(C1,ATL)$

$load(C1,P1,ATL)$

$\neg at(C1,ATL) \wedge in(C1,P1)$

$at(P1,ATL)$

$fly(P1,ATL,MSY)$

$\neg at(P1,ATL) \wedge at(P1,MSY)$

$at(P1,MSY) \wedge in(C1,P1)$

$unload(C1,P1,MSY)$

$at(C1,MSY) \wedge \neg in(C1,P1)$

$at(C1,MSY)$

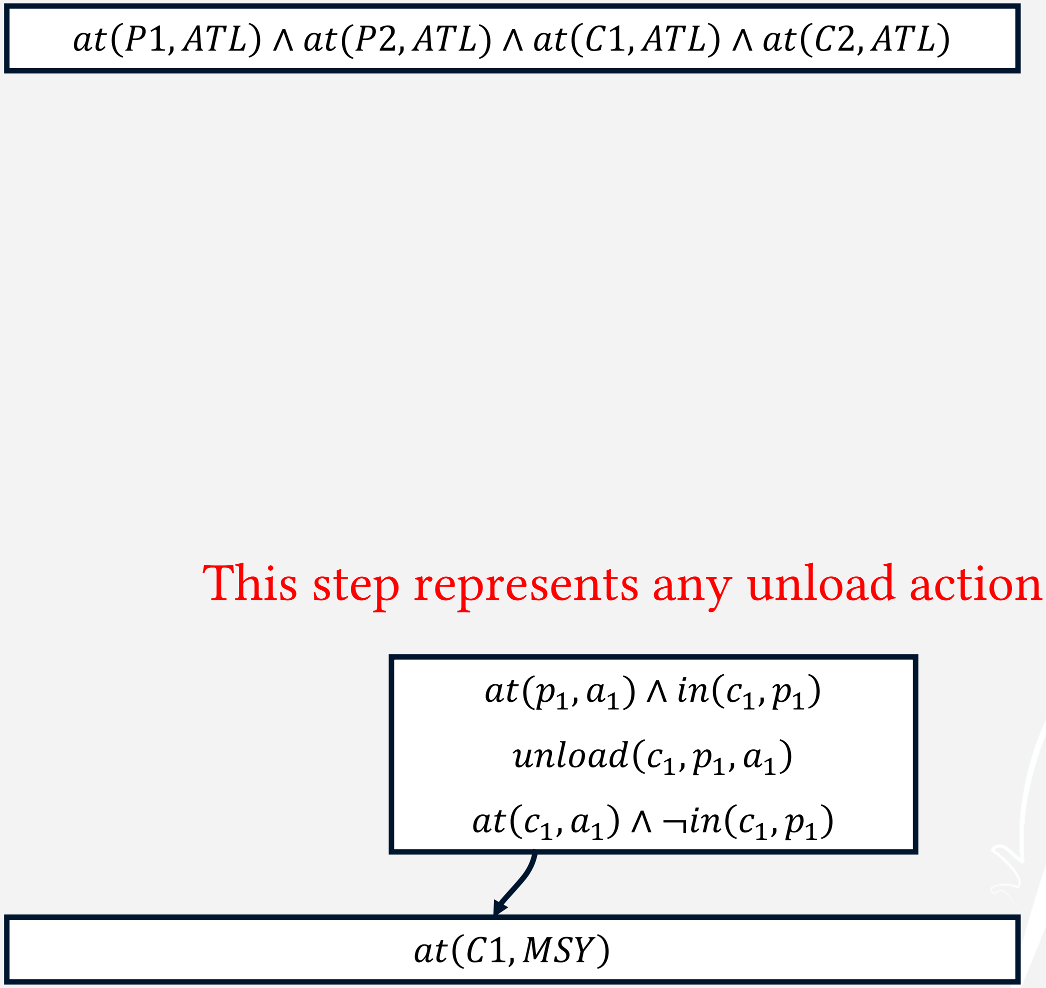
Abstraction

Idea: Even though there are two ways to achieve $at(C1, MSY)$, both are $unload$ actions in which $C1$ is unloaded at MSY. The choice of plane does not matter. Can we use this idea to create a least commitment planner?

Only bind the variables in an action when we know which value they must have.

$$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$$

$$at(C1, MSY)$$



$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

This step represents any unload action.

$at(p_1, a_1) \land in(c_1, p_1)$

$unload(c_1, p_1, a_1)$

$at(c_1, a_1) \land \neg in(c_1, p_1)$

$at(C1, MSY)$

$$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$$

Unify these two expressions.

$$at(p_1, a_1) \land in(c_1, p_1)$$

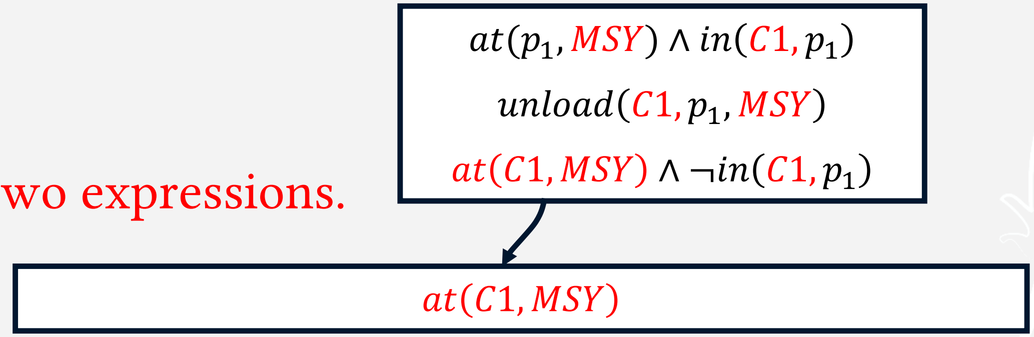
$$unload(c_1, p_1, a_1)$$

$$at(c_1, a_1) \land \neg in(c_1, p_1)$$

$$at(C1, MSY)$$

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

Unify these two expressions.



$at(p_1, MSY) \wedge in(C1, p_1)$

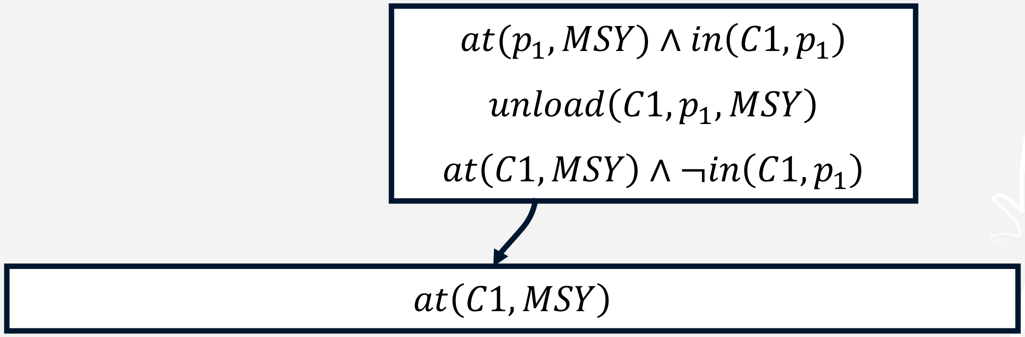
$unload(C1, p_1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, p_1)$

$at(C1, MSY)$

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

This step represents unloading $C1$ from any plane at MSY.

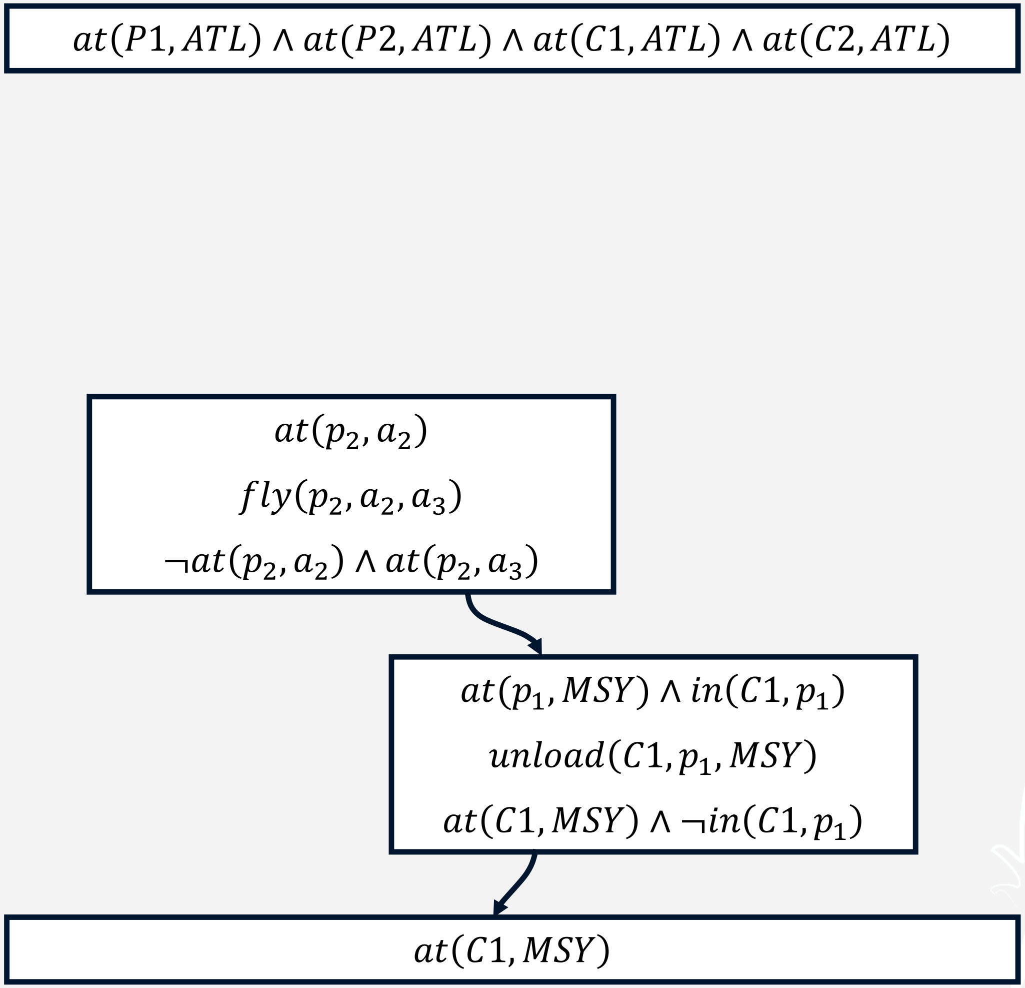


$at(p_1, MSY) \land in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \land \neg in(C1, p_1)$

$at(C1, MSY)$



$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(p_2, a_2)$

$fly(p_2, a_2, a_3)$

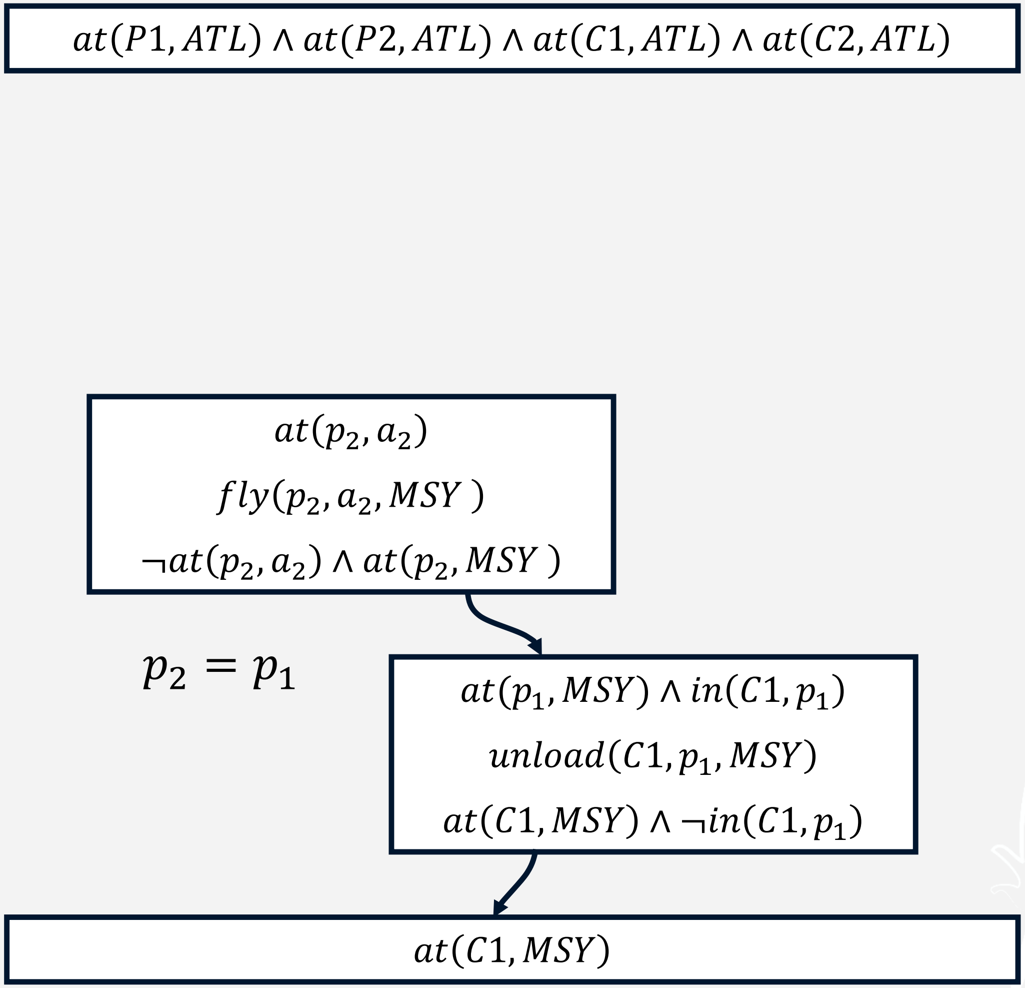
$\neg at(p_2, a_2) \land at(p_2, a_3)$

$at(p_1, MSY) \land in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \land \neg in(C1, p_1)$

$at(C1, MSY)$



$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(p_2, a_2)$

$fly(p_2, a_2, MSY\ )$

$\neg at(p_2, a_2) \land at(p_2, MSY\ )$

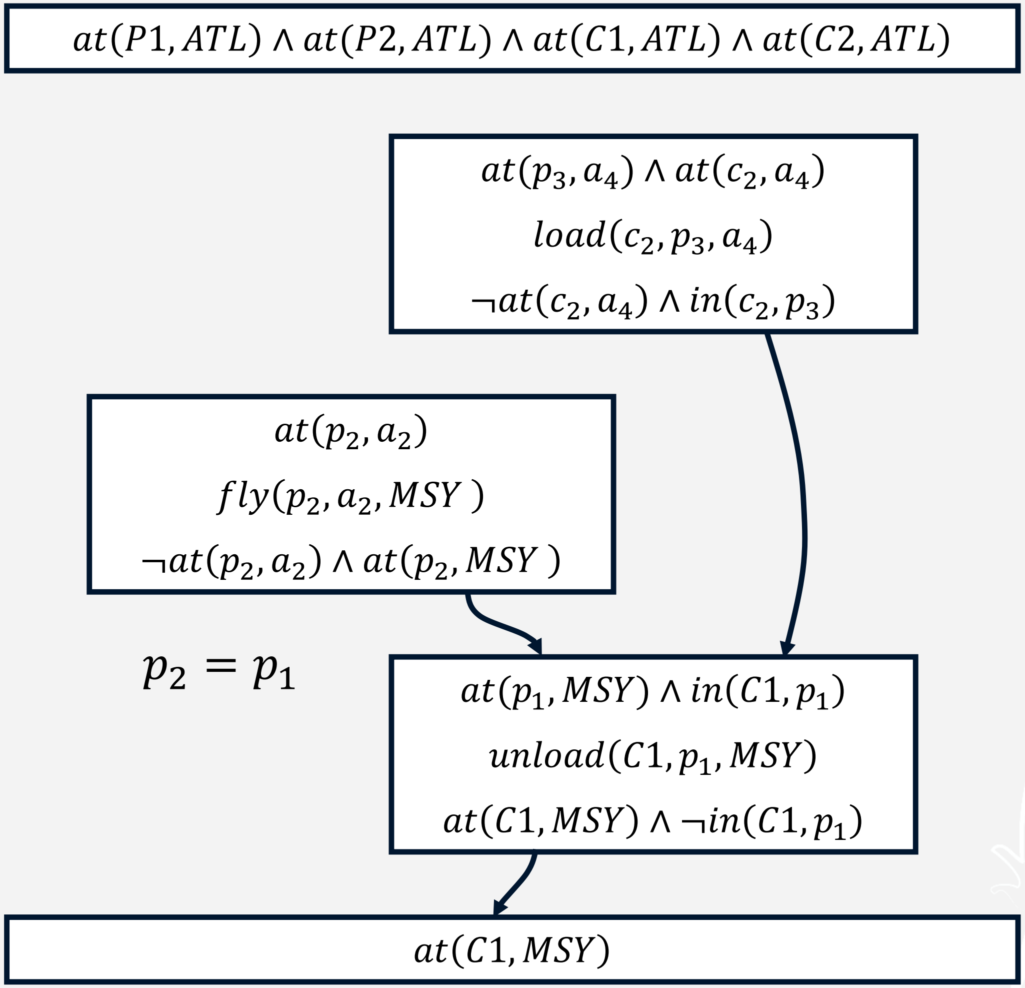
$p_2 = p_1$

$at(p_1, MSY) \land in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \land \neg in(C1, p_1)$

$at(C1, MSY)$



$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$at(p_3, a_4) \wedge at(c_2, a_4)$

$load(c_2, p_3, a_4)$

$\neg at(c_2, a_4) \wedge in(c_2, p_3)$

$at(p_2, a_2)$

$fly(p_2, a_2, MSY)$

$\neg at(p_2, a_2) \wedge at(p_2, MSY)$

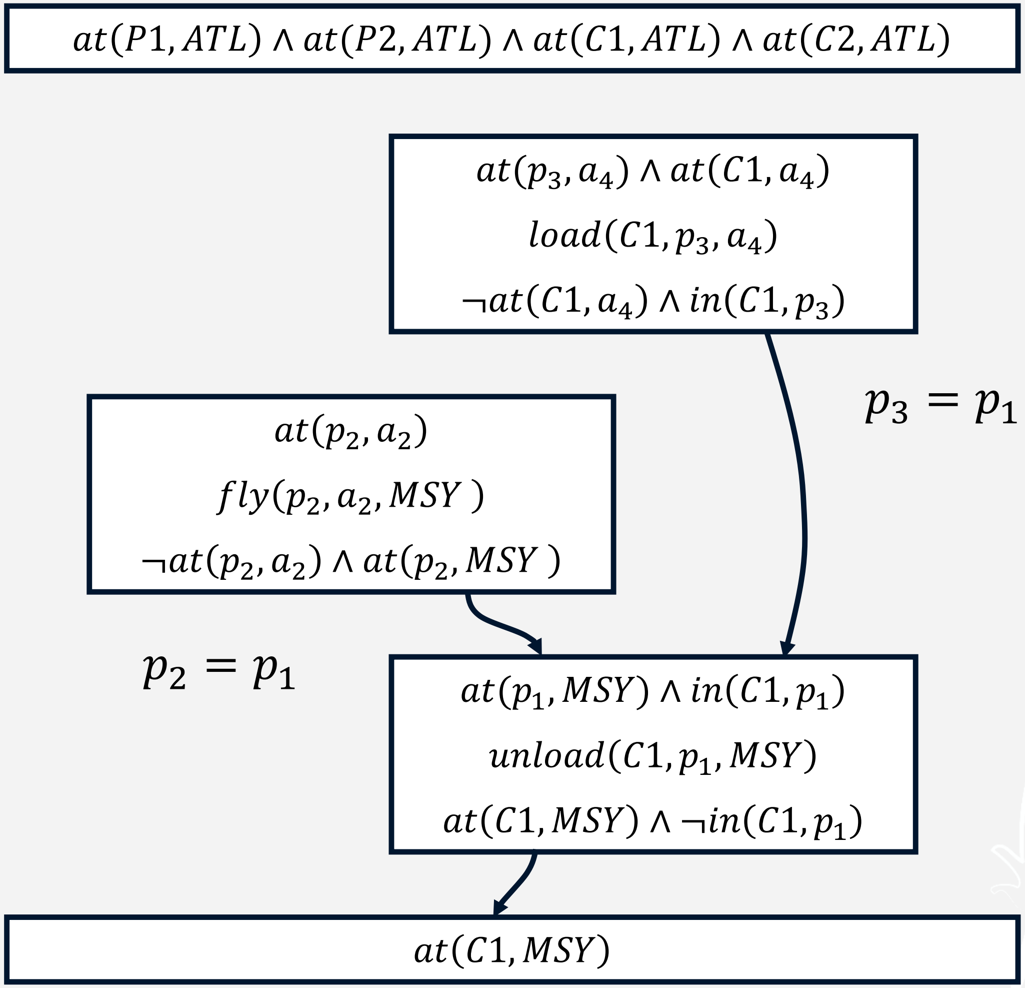
$p_2 = p_1$

$at(p_1, MSY) \wedge in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, p_1)$

$at(C1, MSY)$



$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$at(p_3, a_4) \wedge at(C1, a_4)$

$load(C1, p_3, a_4)$

$\neg at(C1, a_4) \wedge in(C1, p_3)$

$p_3 = p_1$

$at(p_2, a_2)$

$fly(p_2, a_2, MSY\ )$

$\neg at(p_2, a_2) \wedge at(p_2, MSY\ )$

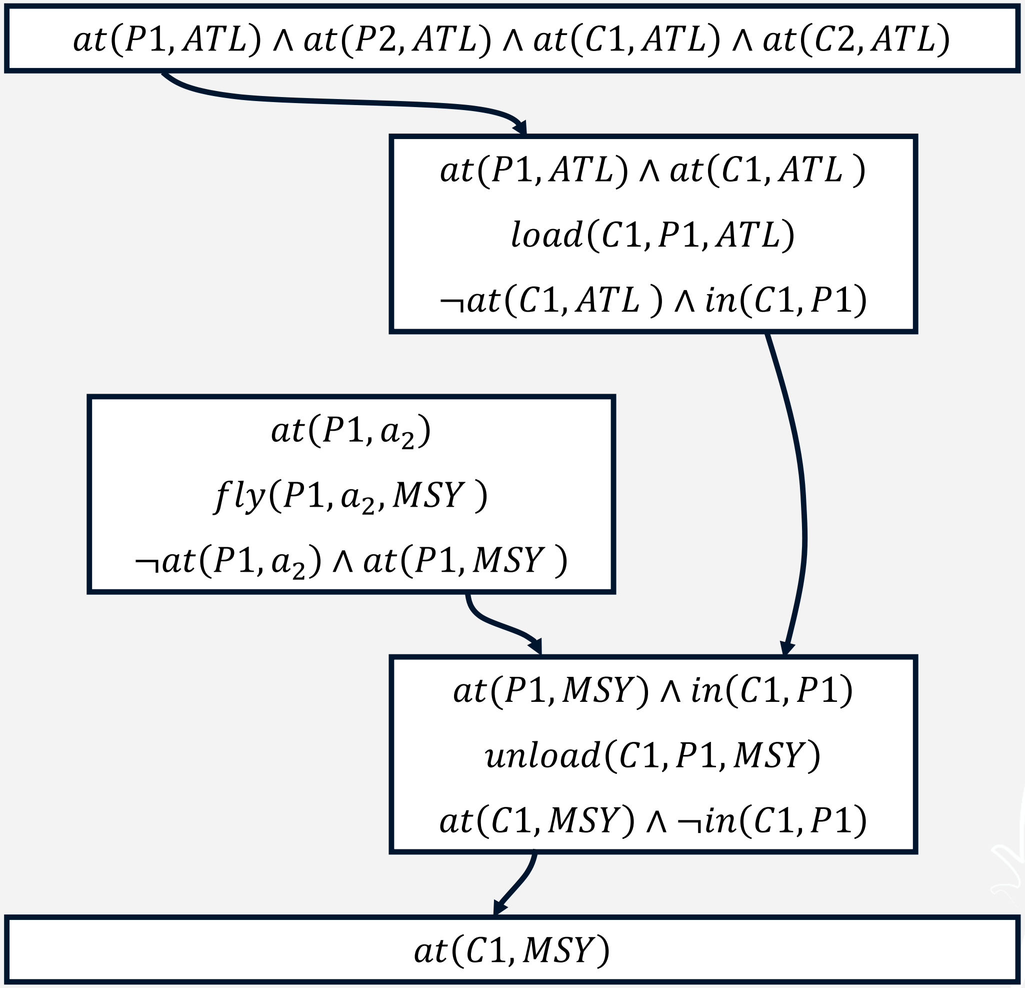
$p_2 = p_1$

$at(p_1, MSY) \wedge in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, p_1)$

$at(C1, MSY)$



$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$at(P1, ATL) \wedge at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \wedge in(C1, P1)$

$at(P1, a_2)$

$fly(P1, a_2, MSY)$

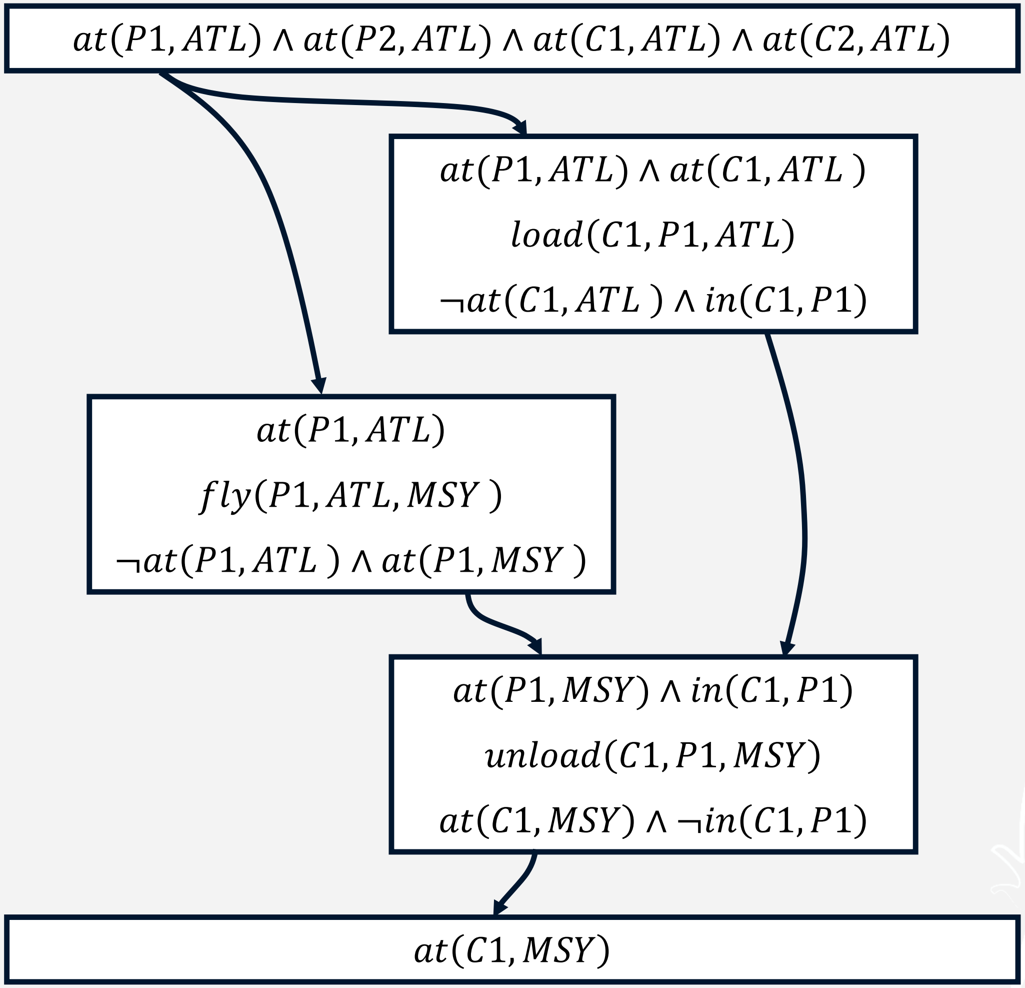
$\neg at(P1, a_2) \wedge at(P1, MSY)$

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$



$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(P1, ATL) \land at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \land in(C1, P1)$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \land at(P1, MSY)$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

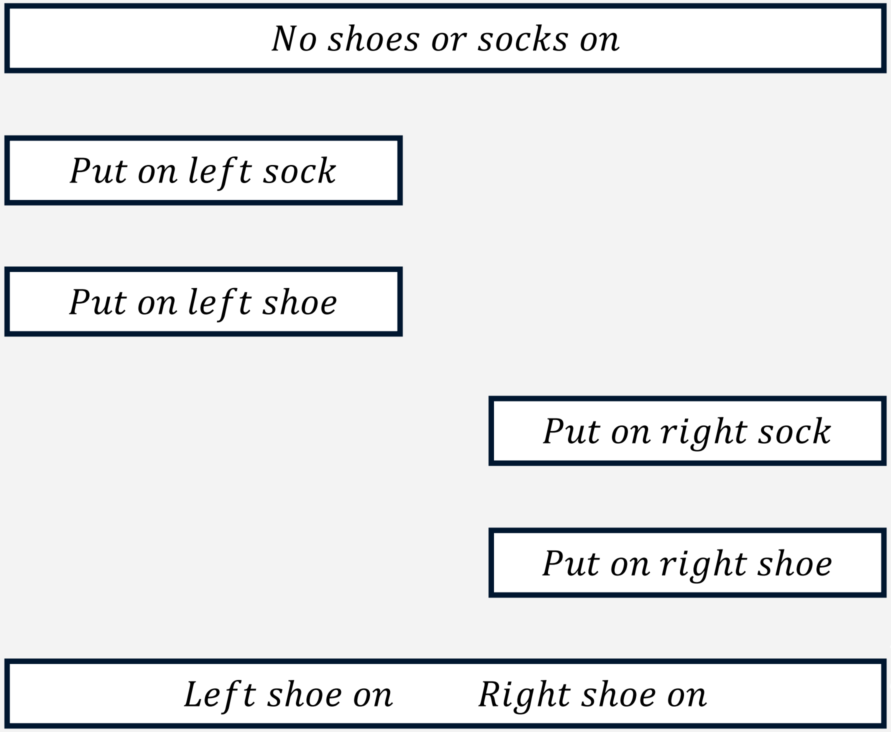
$at(C1, MSY)$

Shoes and Socks Domain

No shoes or socks on

Left shoe on　　*Right shoe on*

Shoes and Socks Domain



No shoes or socks on

Put on left sock

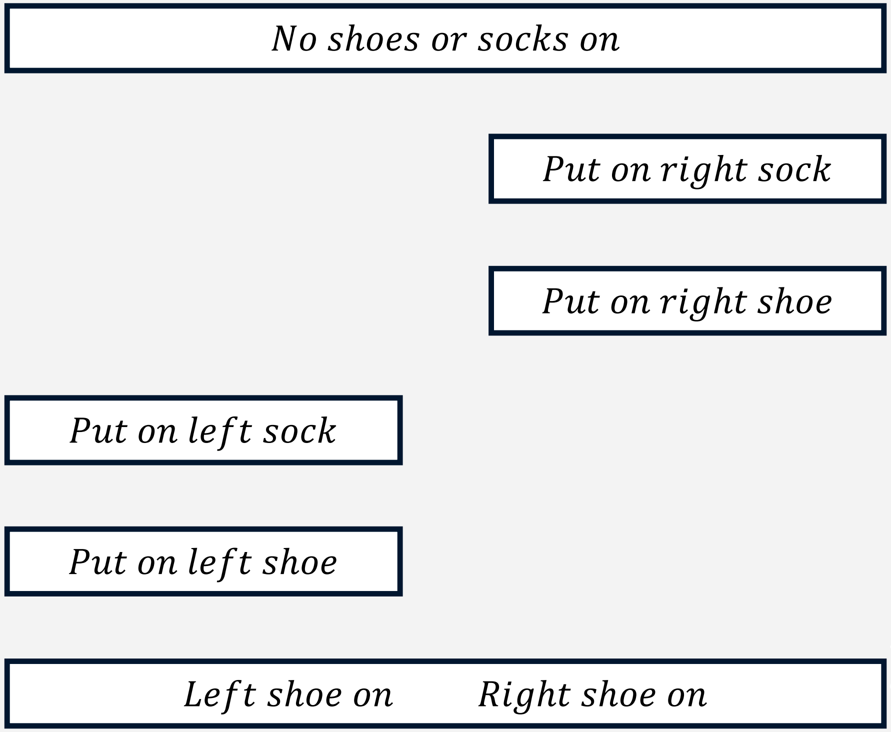
Put on left shoe

Put on right sock

Put on right shoe

Left shoe on   Right shoe on

Shoes and Socks Domain



No shoes or socks on

Put on right sock

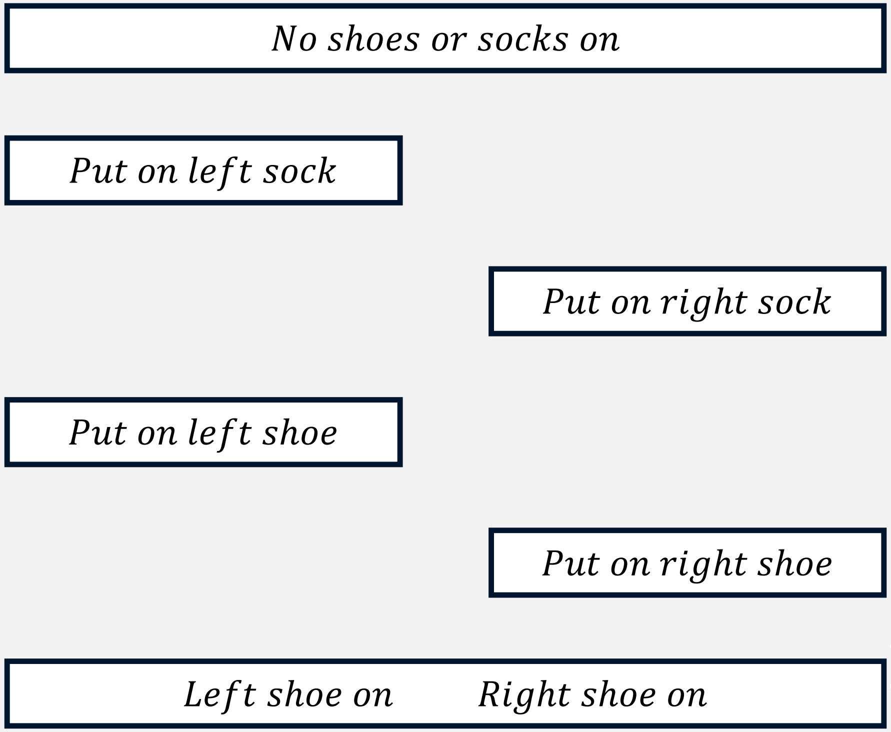
Put on right shoe

Put on left sock

Put on left shoe

Left shoe on *Right shoe on*

Shoes and Socks Domain



No shoes or socks on

Put on left sock

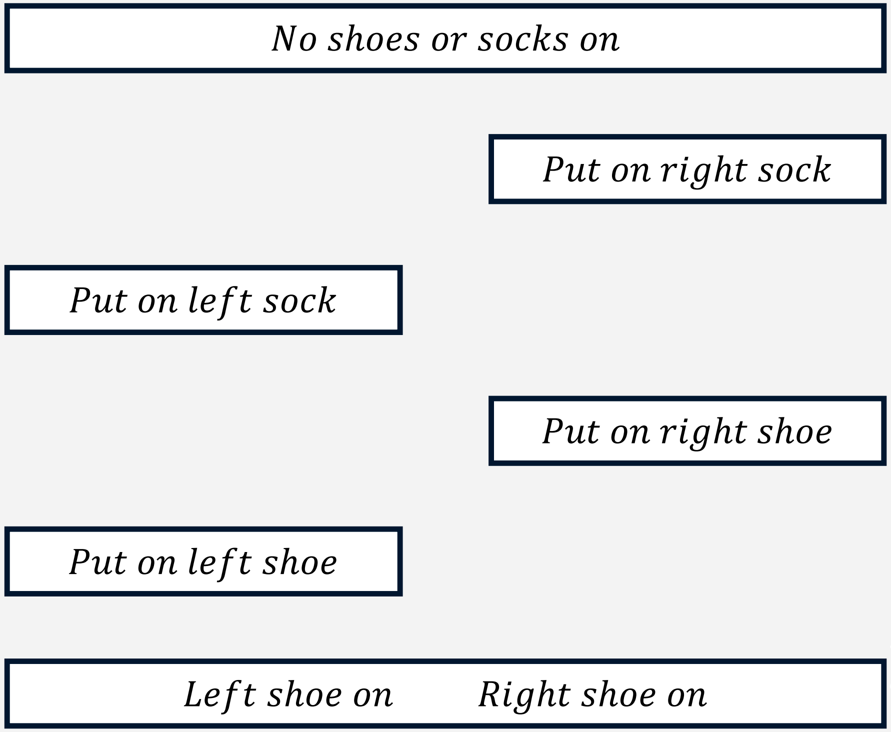
Put on right sock

Put on left shoe

Put on right shoe

Left shoe on Right shoe on

Shoes and Socks Domain



No shoes or socks on

Put on right sock

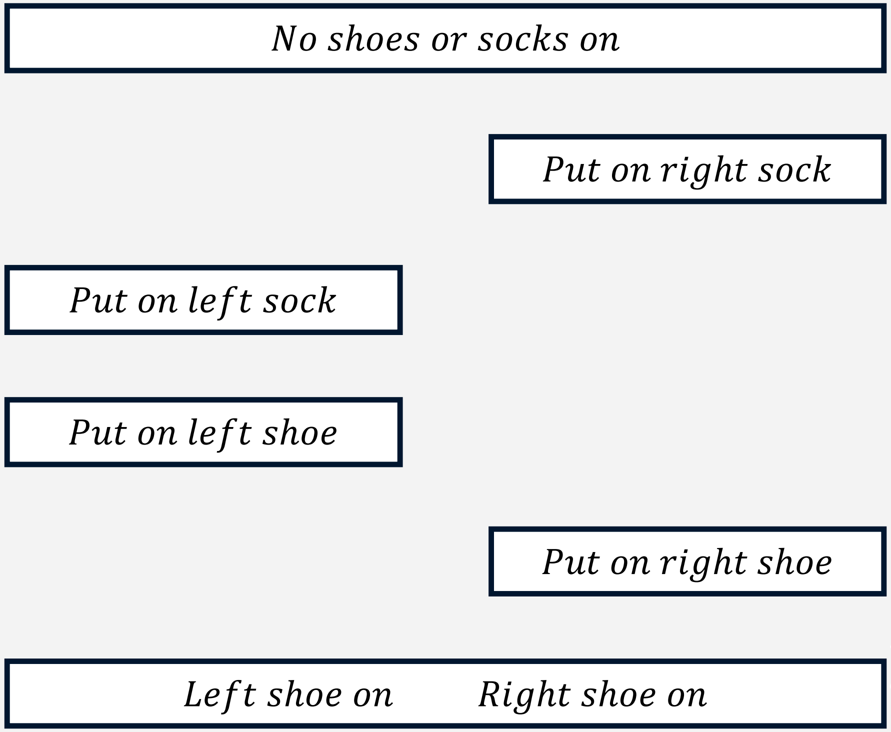
Put on left sock

Put on right shoe

Put on left shoe

Left shoe on Right shoe on

Shoes and Socks Domain



No shoes or socks on

Put on right sock

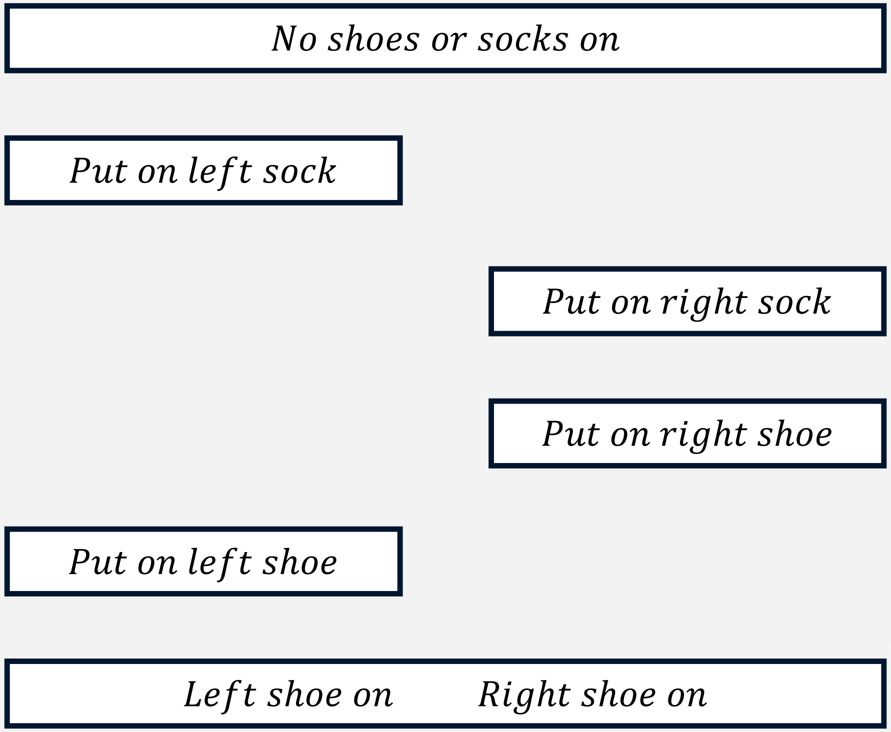
Put on left sock

Put on left shoe

Put on right shoe

Left shoe on *Right shoe on*

Shoes and Socks Domain



No shoes or socks on

Put on left sock

Put on right sock

Put on right shoe

Put on left shoe

Left shoe on Right shoe on

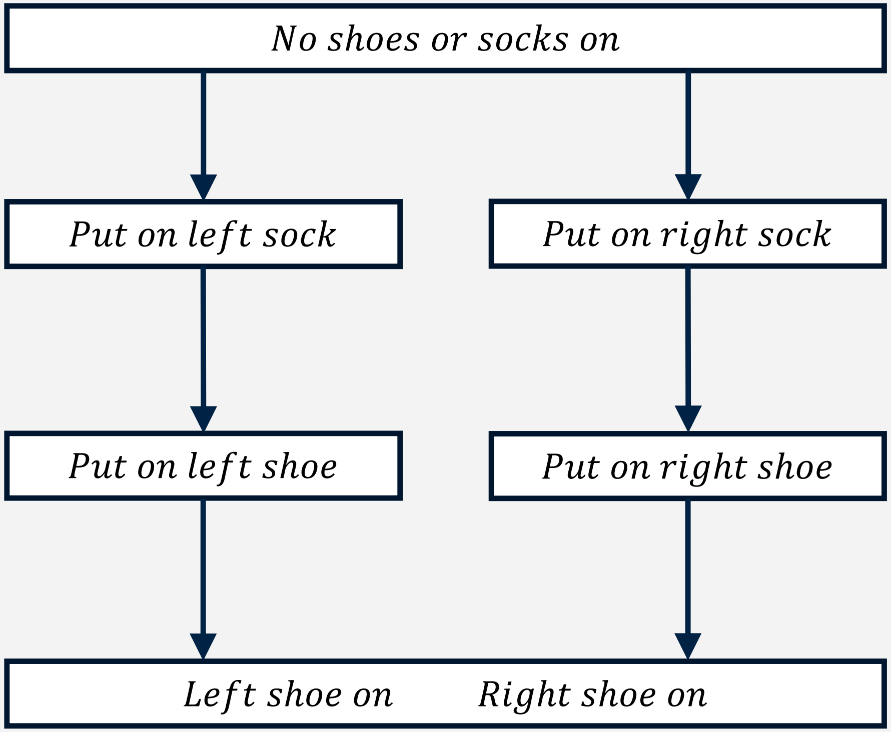
Action Ordering

Idea: There are often many possible orders in which the same step can be taken, but the plans are conceptually similar.

Impose only a partial ordering rather than a total ordering on the steps.

A **partial ordering** is a set of constraints $S1 < S2$ which express the idea that "Step $S1$ must occur before step $S2$, but exactly when does not matter."

Partial Order Plan



Partial Ordering:

Left sock < *Left shoe*

Right sock < *Right shoe*

Implied for all plans:

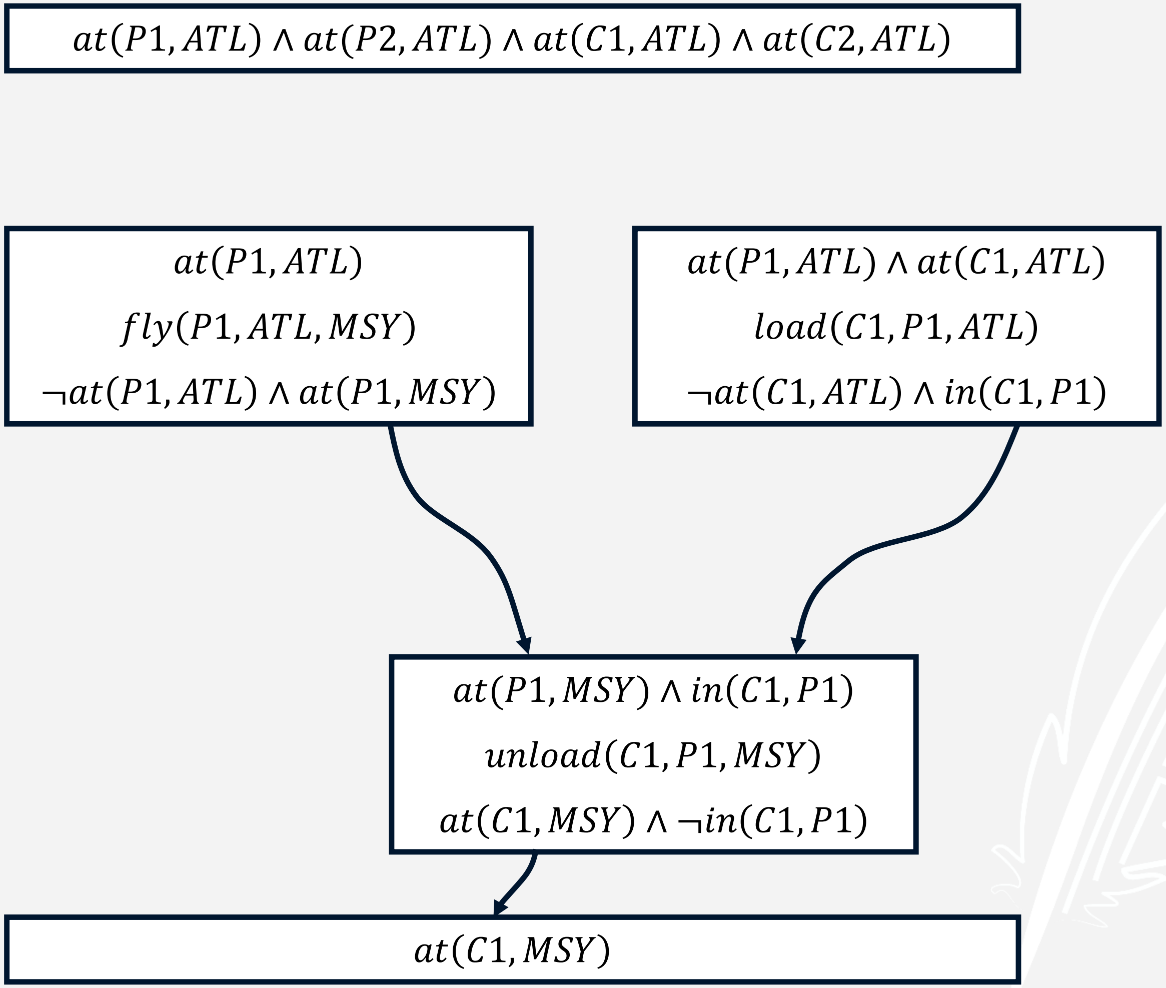
start < *all steps*

all steps < *end*

Interleaved Goals

Partial ordering allows us to separate one goal from another.

Problem: The actions needed to achieve one goal may interfere with the actions needed to achieve a another goal.



$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \land at(P1, MSY)$

$at(P1, ATL) \land at(C1, ATL)$

$load(C1, P1, ATL)$

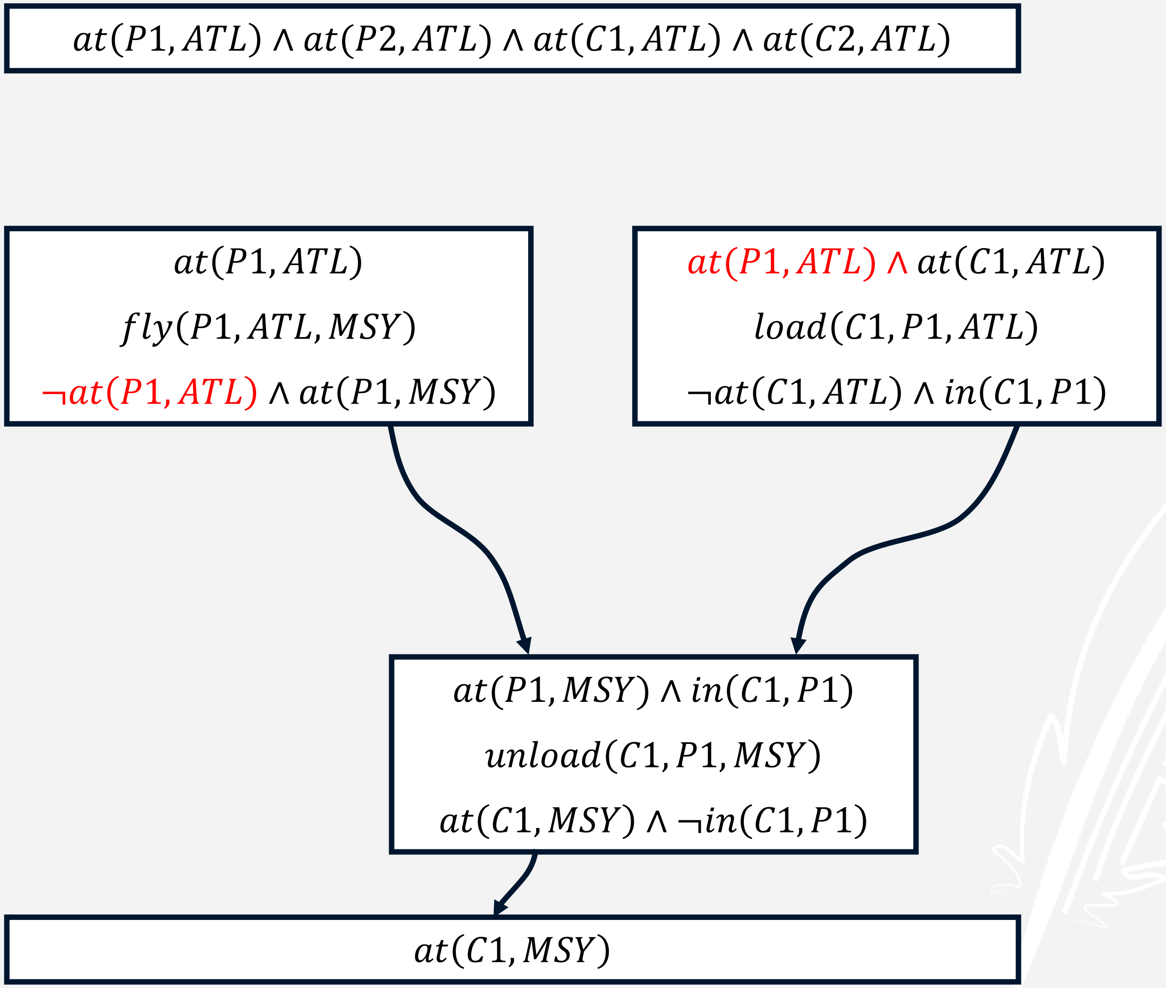
$\neg at(C1, ATL) \land in(C1, P1)$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

$at(C1, MSY)$



$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \land at(P1, MSY)$

$at(P1, ATL) \land at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \land in(C1, P1)$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

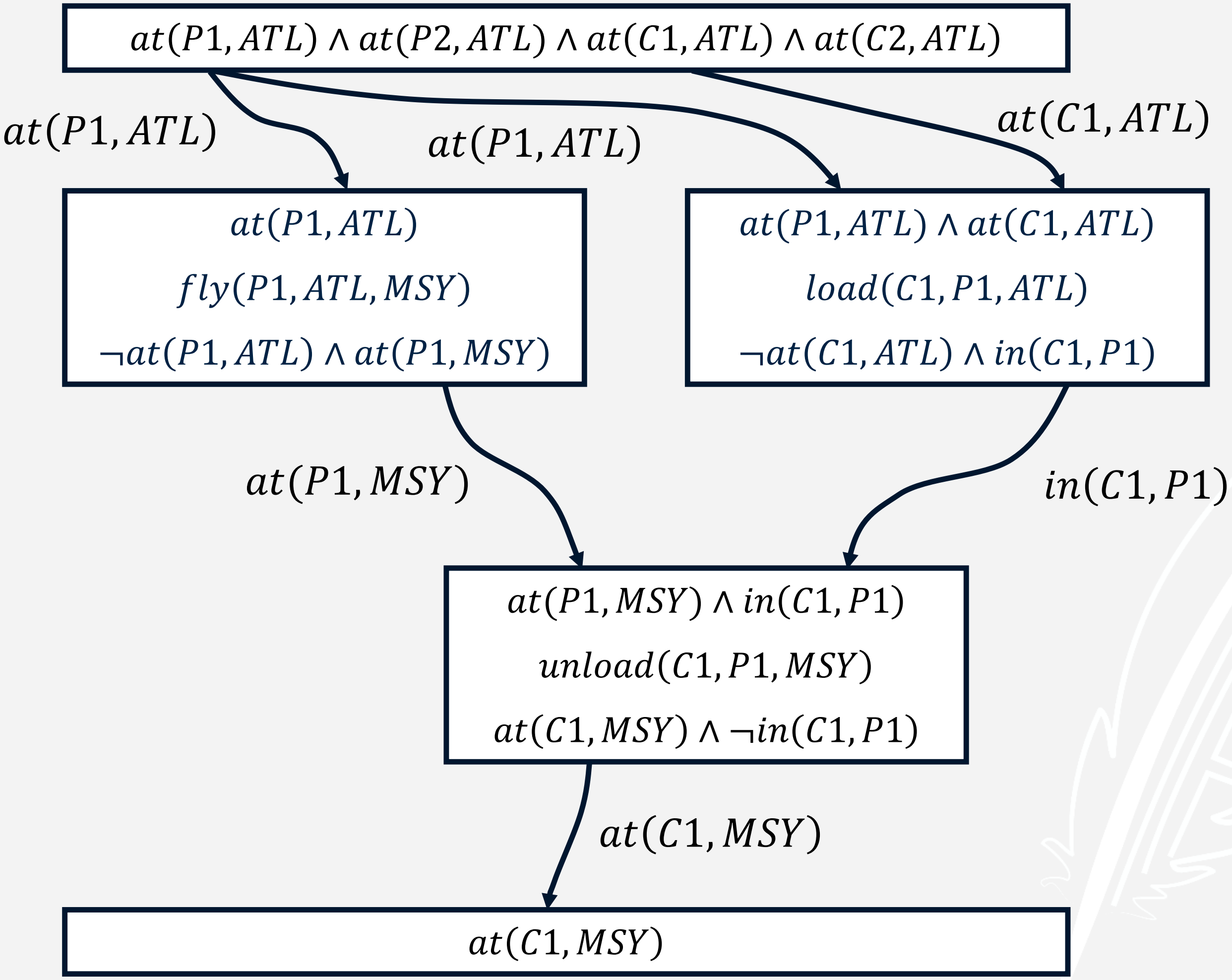
$at(C1, MSY)$

Causal Links

If we consider the plan as a graph, a **causal link** is a directed edge $S1 \xrightarrow{p} S2$ with label p such that:

- The **tail** is a step with effect p.
- The **head** is a step with precondition p.

A causal link implies the ordering $S1 < S2$.



$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$at(P1, ATL)$

$at(P1, ATL)$

$at(C1, ATL)$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

$at(P1, ATL) \wedge at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \wedge in(C1, P1)$

$at(P1, MSY)$

$in(C1, P1)$

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$

$at(C1, MSY)$

Causal Links

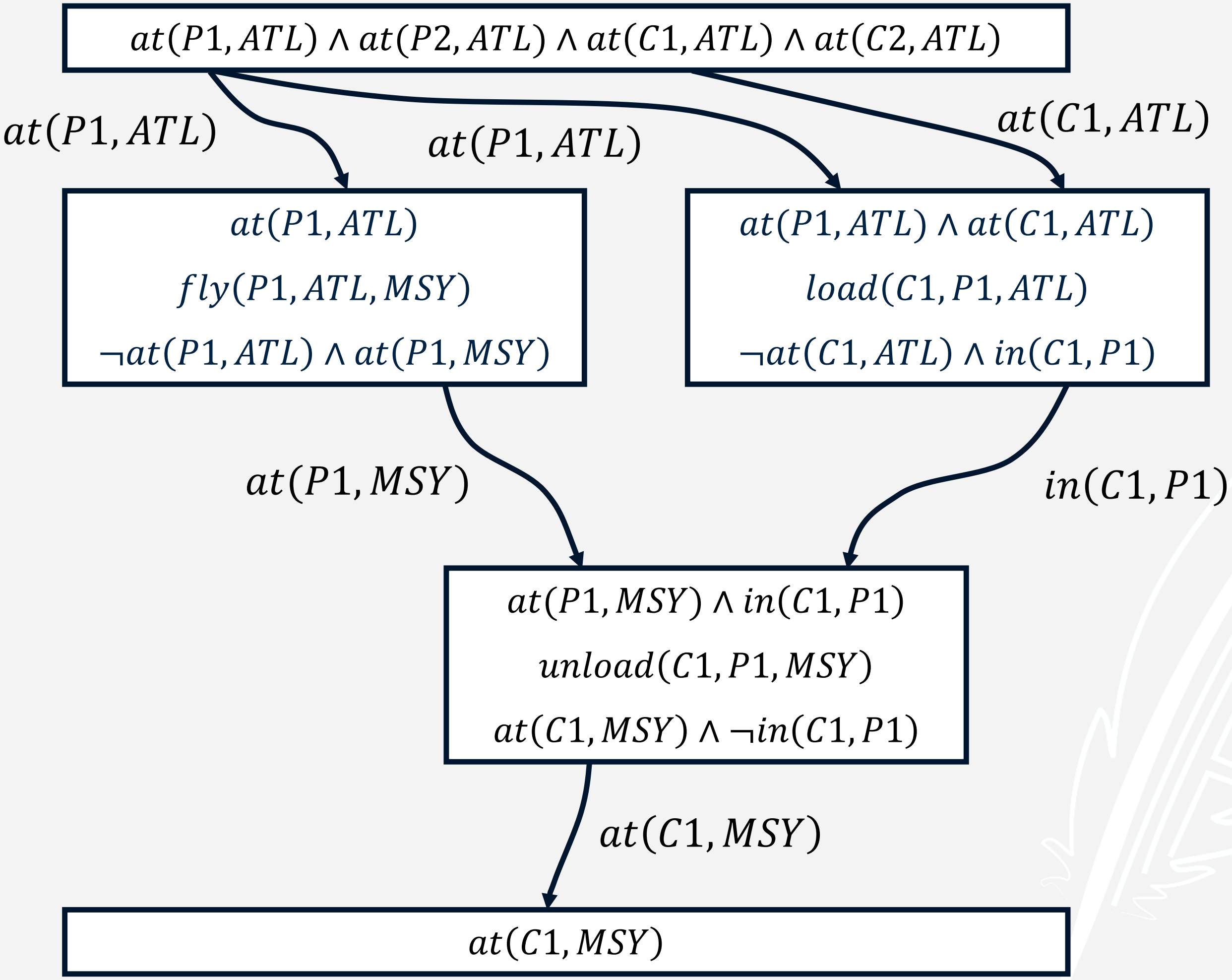
A causal link explains how an earlier step satisfies the precondition of a later step.

The problem with interleaving goals is that the commitment a causal link represents can be undone.

Threatened Causal Links

A causal link $S1 \xrightarrow{p} S2$ is **threatened** by a step T iff:

- Step T has effect $\neg p$.
- The current partial ordering allows $S1 < T < S2$.



$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$at(P1, ATL)$

$at(P1, ATL)$

$at(C1, ATL)$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

$at(P1, ATL) \wedge at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \wedge in(C1, P1)$

$at(P1, MSY)$

$in(C1, P1)$

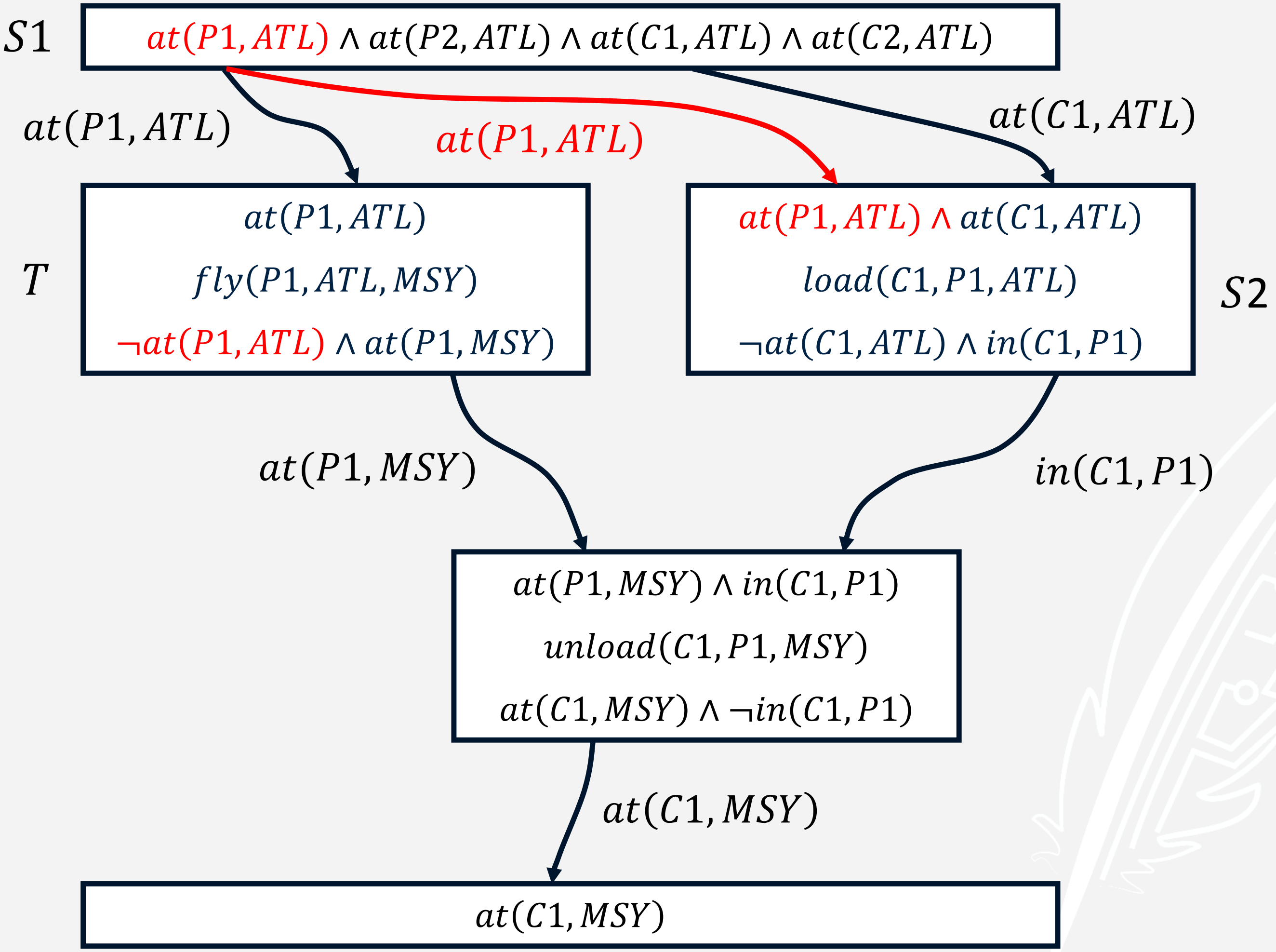
$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$

$at(C1, MSY)$



$S1$

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$at(P1, ATL)$

$at(P1, ATL)$

$at(C1, ATL)$

T

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

$at(P1, ATL) \wedge at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \wedge in(C1, P1)$

$S2$

$at(P1, MSY)$

$in(C1, P1)$

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$

$at(C1, MSY)$

Fixing Threatened Causal Links

When a causal link is threatened, it means that a step exists which could occur between the tail and the head which undoes the fact established by the link.

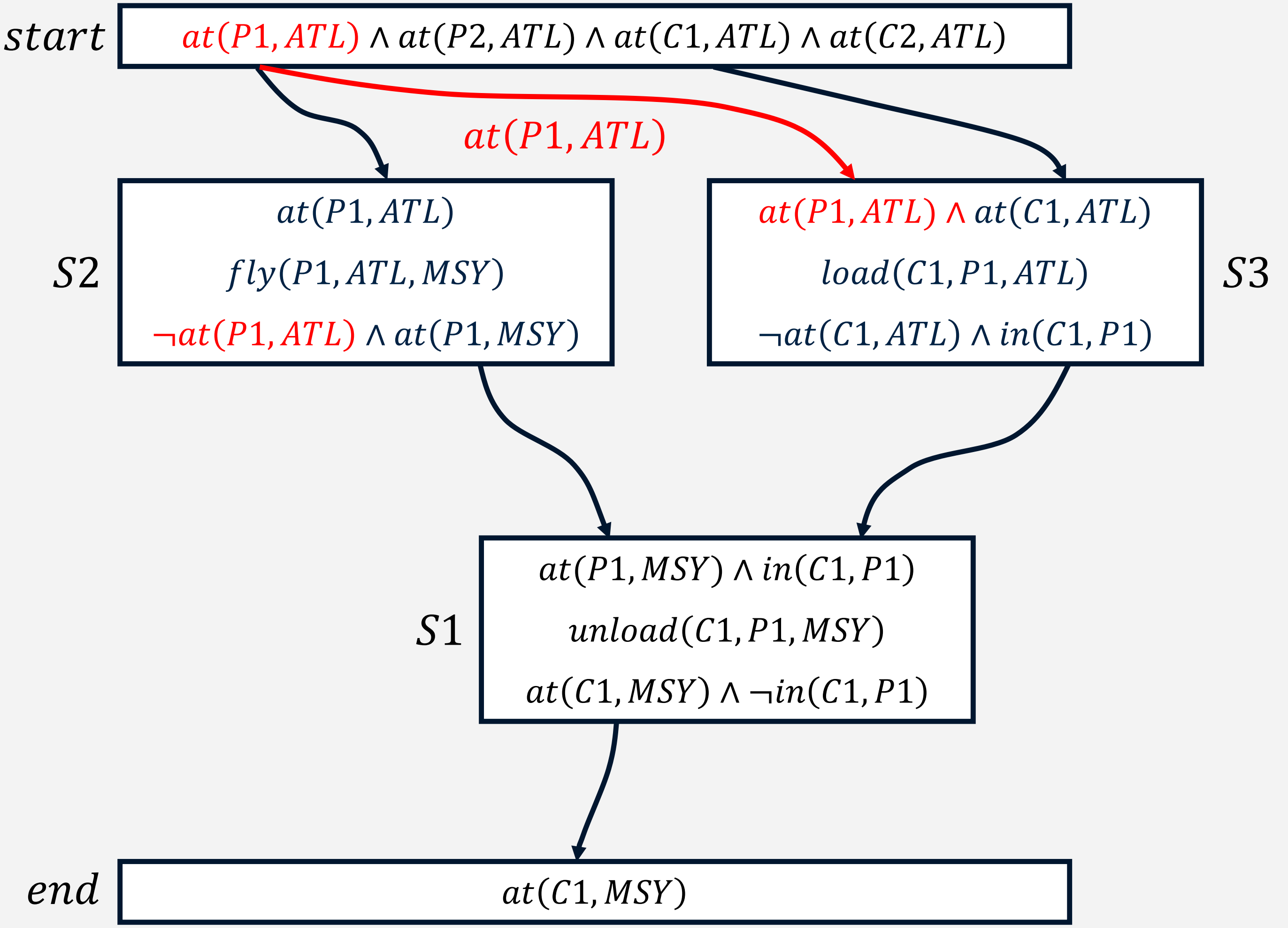
Solution: We can fix threatened causal links by adding new orderings to the plan which ensure the link is not threatened.

Fixing Threatened Causal Links

Given a causal link $S1 \xrightarrow{p} S2$ and a threatening step T, we can remove the threat in two ways:

- **Promotion**: order $S2 < T$.
- **Demotion**: order $T < S1$.

Note: These can only be done if it does not make the partial ordering impossible (i.e. does not create a cycle in the orderings).



$start$ $at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(P1, ATL)$

$S2$ $at(P1, ATL)$ $fly(P1, ATL, MSY)$ $\neg at(P1, ATL) \land at(P1, MSY)$

$S3$ $at(P1, ATL) \land at(C1, ATL)$ $load(C1, P1, ATL)$ $\neg at(C1, ATL) \land in(C1, P1)$

$S1$ $at(P1, MSY) \land in(C1, P1)$ $unload(C1, P1, MSY)$ $at(C1, MSY) \land \neg in(C1, P1)$

end $at(C1, MSY)$

Orderings:

$start < end$

$start < S1$

$S1 < end$

$start < S2$

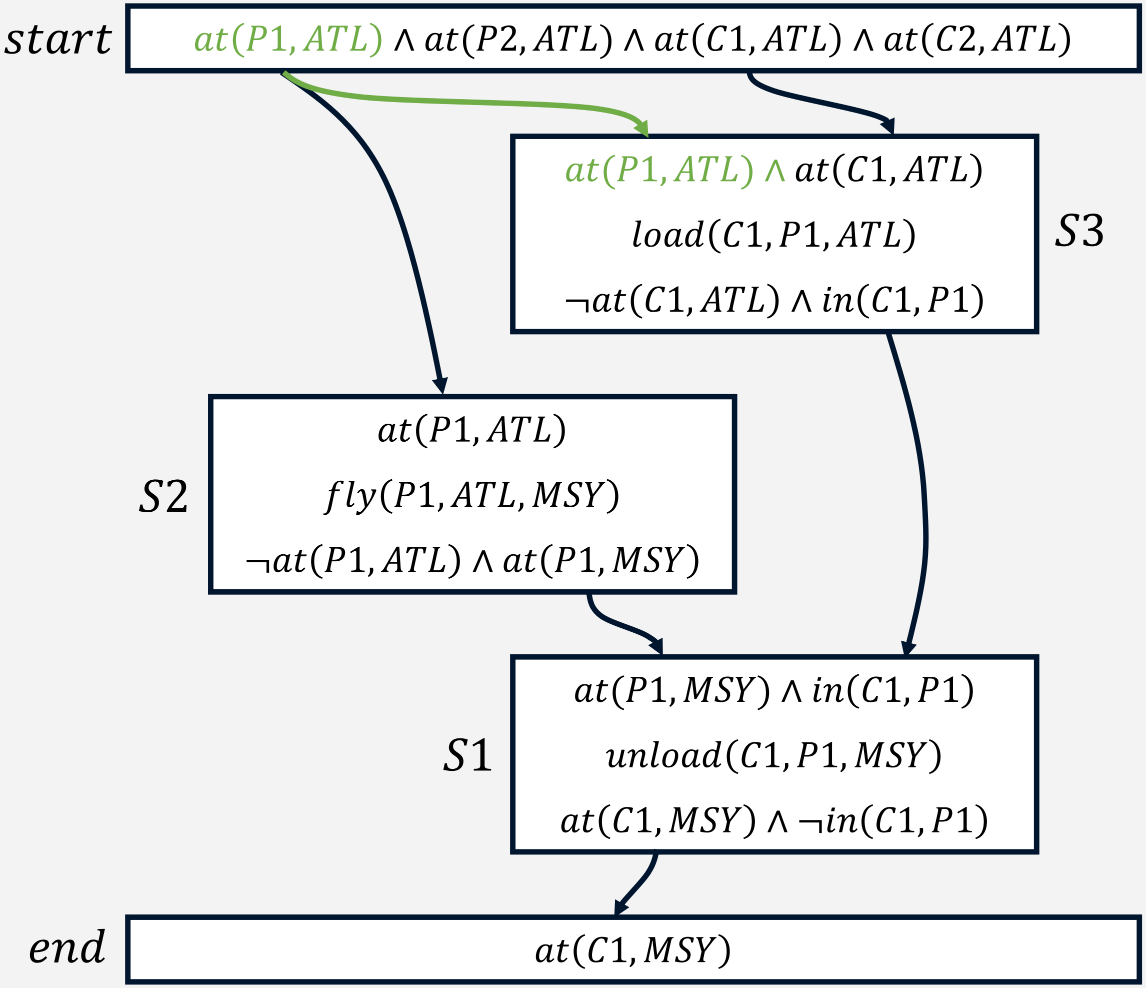
$S2 < end$

$S2 < S1$

$start < S3$

$S3 < end$

$S3 < S1$



start $at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$at(P1, ATL) \wedge at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \wedge in(C1, P1)$

S3

S2

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

S1

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

end $at(C1, MSY)$

Orderings:

$start < end$

$start < S1$

$S1 < end$

$start < S2$

$S2 < end$

$S2 < S1$

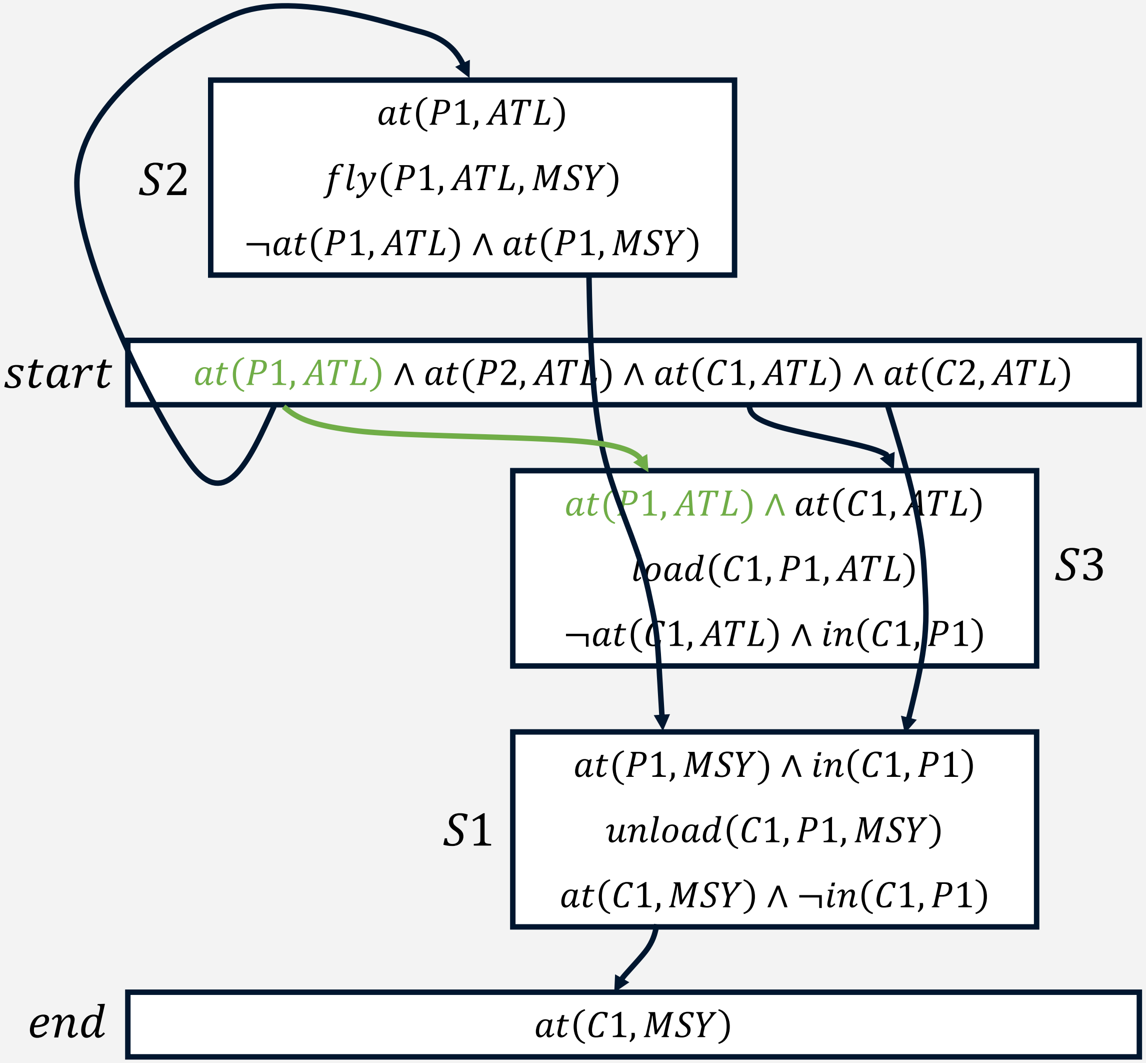
$start < S3$

$S3 < end$

$S3 < S1$

$S3 < S2$

Promote



$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

$start$

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$at(P1, ATL) \wedge at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \wedge in(C1, P1)$

$S3$

$S1$

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

end

$at(C1, MSY)$

Orderings:

$start < end$

$start < S1$

$S1 < end$

$start < S2$

$S2 < end$

$S2 < S1$

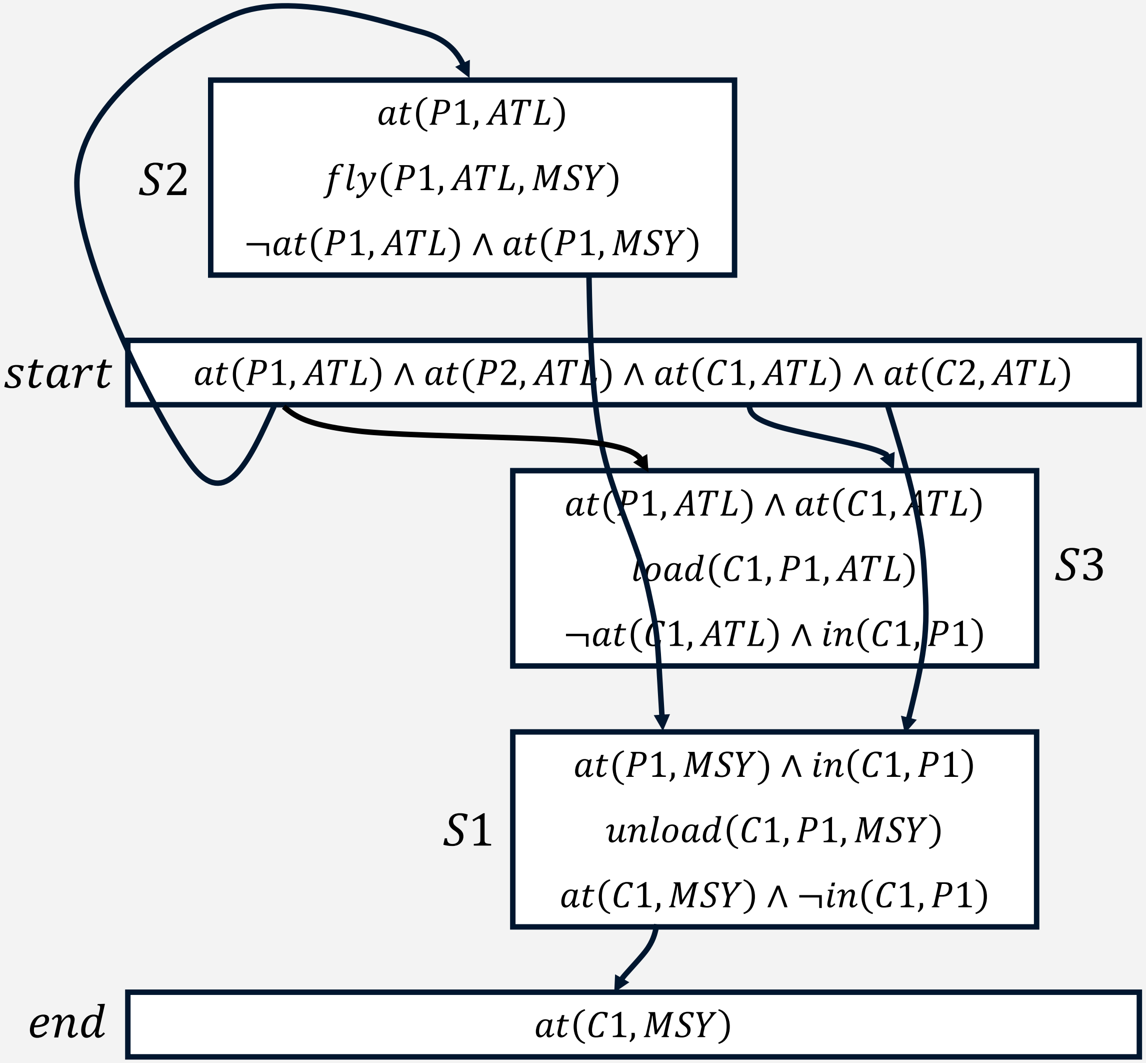
$start < S3$

$S3 < end$

$S3 < S1$

$S2 < start$

Demote


S2
$at(P1, ATL)$
$fly(P1, ATL, MSY)$
$\neg at(P1, ATL) \wedge at(P1, MSY)$
start
$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$
$at(P1, ATL) \wedge at(C1, ATL)$
$load(C1, P1, ATL)$
$\neg at(C1, ATL) \wedge in(C1, P1)$
S3
S1
$at(P1, MSY) \wedge in(C1, P1)$
$unload(C1, P1, MSY)$
$at(C1, MSY) \wedge \neg in(C1, P1)$
end
$at(C1, MSY)$

Orderings:

$start < end$

$start < S1$

$S1 < end$

$start < S2$

$S2 < end$

$S2 < S1$

$start < S3$

$S3 < end$

$S3 < S1$

$S2 < start$

Demote fails!

Least Commitment Planning

So far, we have explored three ways to reduce the size of the search space:

- Backwards search often lowers the branching factor by considering only relevant steps.
- Leaving some variables unbound in a step allows it to represent many possible steps at once.
- A partial ordering can represent many possible total orderings (as long as we ensure that no causal links are threatened).

Least Commitment Planning

The problem with building plans in a least-commitment fashion is that the idea of the *current state* no longer exists.

- When a plan is built backwards, we don't know what the early steps are until the end.
- When a step has unbound variables, we don't know exactly what its effects are.
- We don't know the exact order of steps in a partial order plan, so we don't know the current state.

State-Space Search

The most straight-forward way to view planning as search is **state-space search**:

- **State:** A literal or conjunction of literals.
- **Action:** Take an action whose precondition is met and modify the current state according to effects.
- **Goal:** Done when the goal holds in the current state.

We call this state-space search because the nodes in the search space are states.

Plan-Space Search

To take advantage of least-commitment planning, we need to search the space of plans:

- **State:** A (possibly incomplete) partial order plan.
- **Action:** Modifying the plan by adding steps, orderings, or causal links.
- **Goal:** Done when all the goal and preconditions have been satisfied by causal links.

Because the nodes in the search space are plans, we call this **plan-space search**.

Refinement Search

POCL (partial order causal link) planning is a kind of **refinement search**.

A plan is a data structure. If that data structure is incomplete, it has a set of flaws which describe how it is incomplete.

Search proceeds by choosing a flaw and fixing it (possibly creating new flaws in the process).

Search is done when no flaws remain.

POCL Plan Data Structure

A plan is composed of four sets:

- A set of **steps**. Some variables may not be bound.
- A set of **bindings** which constrain which values the variables can have.
- A set of **orderings** which define a partial ordering of the steps.
- A set of **causal links** which keep track of how goals are achieved.

POCL Plan Flaws

An **open precondition flaw** for some step S with precondition p indicates that there is no causal link which established p.

A **threatened causal link flaw** indicates that a threatened causal link exists.

POP Algorithm

```
Begin with the null plan and empty set of flaws F.
For each goal conjunct, add an open precondition flaw to F.
To refine a plan with flaws:
    If the plan has no flaws, return it as a solution.
    Choose a flaw X from F to repair.
    If X is an open precondition flaw for literal L of step S:
        Choose some action A which has L as an effect:
            A can be a step already in the plan, or
            A can be a new step (Add open precondition flaws for A’s preconditions to F.)
        Add a causal link from A to S with label L.
        Add any new threatened causal link flaws to F.
    If X is a threatened causal link flaw:
        Promote: Move the threatening step after the tail.
        Demote: Move the threatening step before the head.
    Recursively repair the refined plan.
```

POP Algorithm

```
Begin with the null plan and empty set of flaws F.
For each goal conjunct, add an open precondition flaw to F.
To refine a plan with flaws:
    If the plan has no flaws, return it as a solution.
    Choose a flaw X from F to repair.
    If X is an open precondition flaw for literal L of step S:
        Choose some action A which has L as an effect:
            A can be a step already in the plan, or
            A can be a new step (Add open precondition flaws for A’s preconditions to F.)
        Add a causal link from A to S with label L.
        Add any new threatened causal link flaws to F.
    If X is a threatened causal link flaw:
        Promote: Move the threatening step after the tail.
        Demote: Move the threatening step before the head.
    Recursively repair the refined plan.
```

POP Algorithm

```
Begin with the null plan and empty set of flaws F.
For each goal conjunct, add an open precondition flaw to F.
To refine a plan with flaws:
    If the plan has no flaws, return it as a solution.
    Choose a flaw X from F to repair.
    If X is an open precondition flaw for literal L of step S:
        Choose some action A which has L as an effect:
            A can be a step already in the plan, or
            A can be a new step (Add open precondition flaws for A's preconditions to F.)
        Add a causal link from A to S with label L.
        Add any new threatened causal link flaws to F.
    If X is a threatened causal link flaw:
        Promote: Move the threatening step after the tail.
        Demote: Move the threatening step before the head.
    Recursively repair the refined plan.
```

POP Algorithm

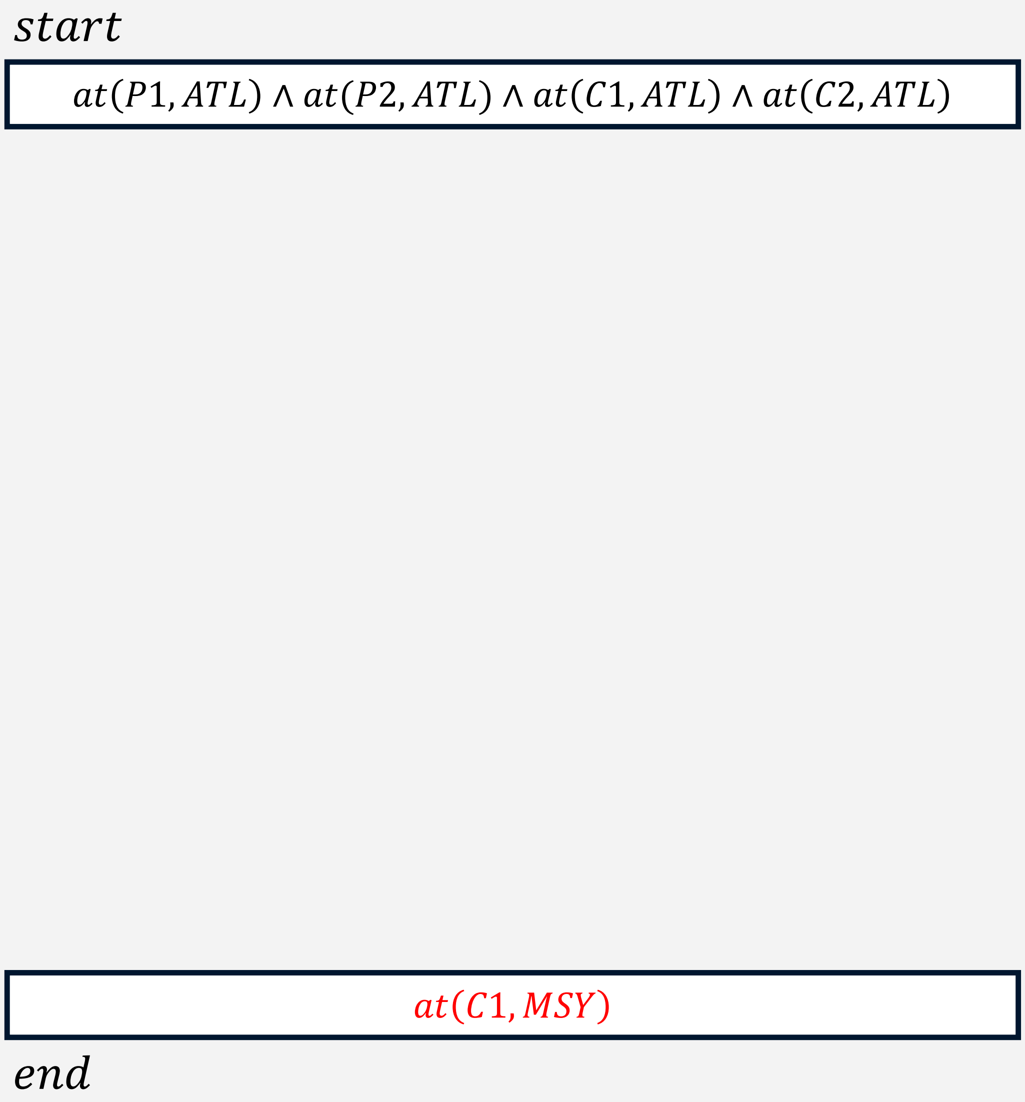
```
Begin with the null plan and empty set of flaws F.
For each goal conjunct, add an open precondition flaw to F.
To refine a plan with flaws:
    If the plan has no flaws, return it as a solution.
    Choose a flaw X from F to repair.
    If X is an open precondition flaw for literal L of step S:
        Choose some action A which has L as an effect:
            A can be a step already in the plan, or
            A can be a new step (Add open precondition flaws for A’s preconditions to F.)
        Add a causal link from A to S with label L.
        Add any new threatened causal link flaws to F.
    If X is a threatened causal link flaw:
        Promote: Move the threatening step after the tail.
        Demote: Move the threatening step before the head.
    Recursively repair the refined plan.
```

POP Algorithm

```
Begin with the null plan and empty set of flaws F.
For each goal conjunct, add an open precondition flaw to F.
To refine a plan with flaws:
    If the plan has no flaws, return it as a solution.
    Choose a flaw X from F to repair.
    If X is an open precondition flaw for literal L of step S:
        Choose some action A which has L as an effect:
            A can be a step already in the plan, or
            A can be a new step (Add open precondition flaws for A's preconditions to F.)
        Add a causal link from A to S with label L.
        Add any new threatened causal link flaws to F.
    If X is a threatened causal link flaw:
        Promote: Move the threatening step after the tail.
        Demote: Move the threatening step before the head.
    Recursively repair the refined plan.
```

POP Algorithm

```
Begin with the null plan and empty set of flaws F.
For each goal conjunct, add an open precondition flaw to F.
To refine a plan with flaws:
    If the plan has no flaws, return it as a solution.
    Choose a flaw X from F to repair.
    If X is an open precondition flaw for literal L of step S:
        Choose some action A which has L as an effect:
            A can be a step already in the plan, or
            A can be a new step (Add open precondition flaws for A’s preconditions to F.)
        Add a causal link from A to S with label L.
        Add any new threatened causal link flaws to F.
    If X is a threatened causal link flaw:
        Promote: Move the threatening step after the tail.
        Demote: Move the threatening step before the head.
    Recursively repair the refined plan.
```



start

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(C1, MSY)$

end

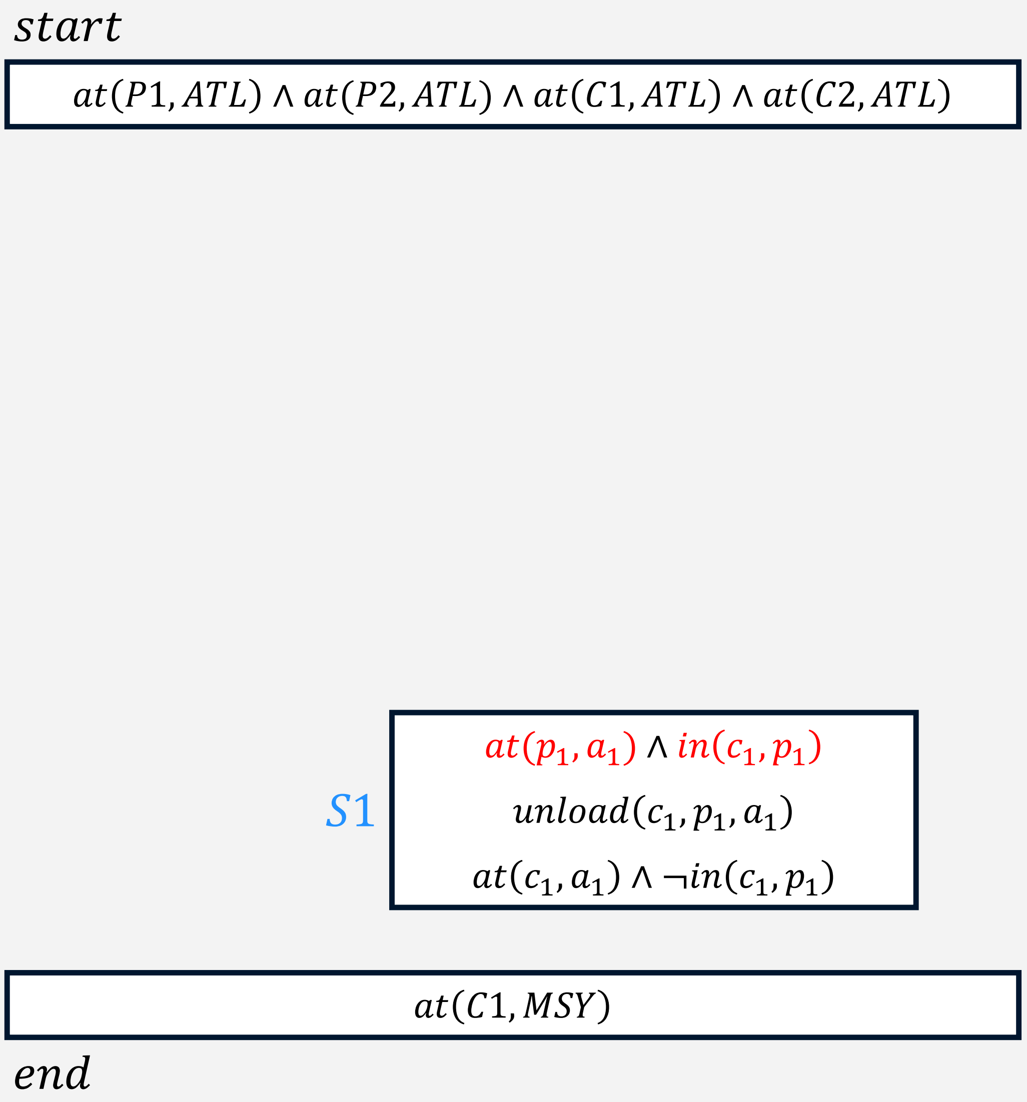
Flaw: $at(C1, MSY)$ open

Steps: $start$, end

Bindings:

Orderings: $start < end$,

Causal Links:



start

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$S1$

$at(p_1, a_1) \wedge in(c_1, p_1)$

$unload(c_1, p_1, a_1)$

$at(c_1, a_1) \wedge \neg in(c_1, p_1)$

$at(C1, MSY)$

end

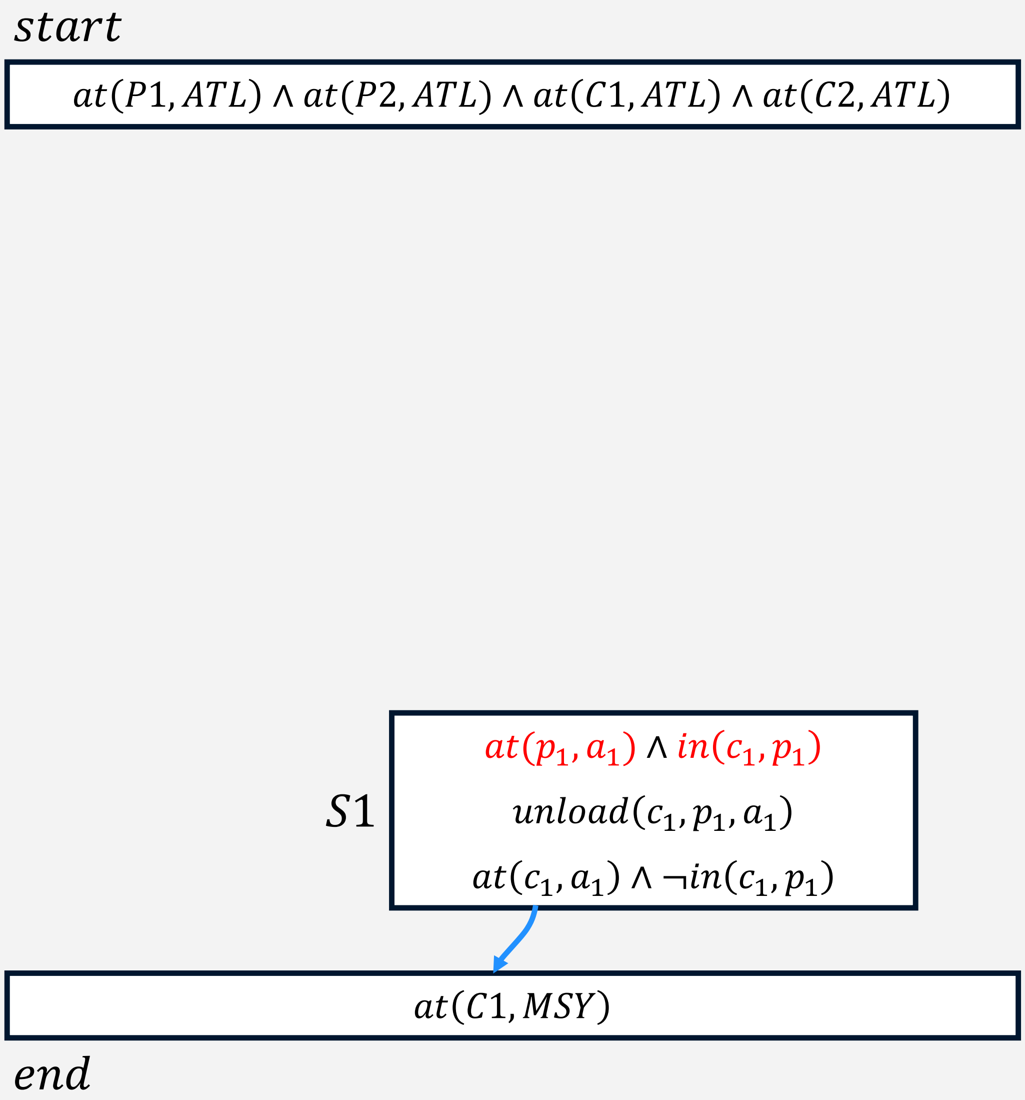
Flaw: $at(C1, MSY)$ open

Steps: $start$, end, $S1$

Bindings:

Orderings: $start < end$

Causal Links:



start

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$S1$

$at(p_1, a_1) \land in(c_1, p_1)$

$unload(c_1, p_1, a_1)$

$at(c_1, a_1) \land \neg in(c_1, p_1)$

$at(C1, MSY)$

end

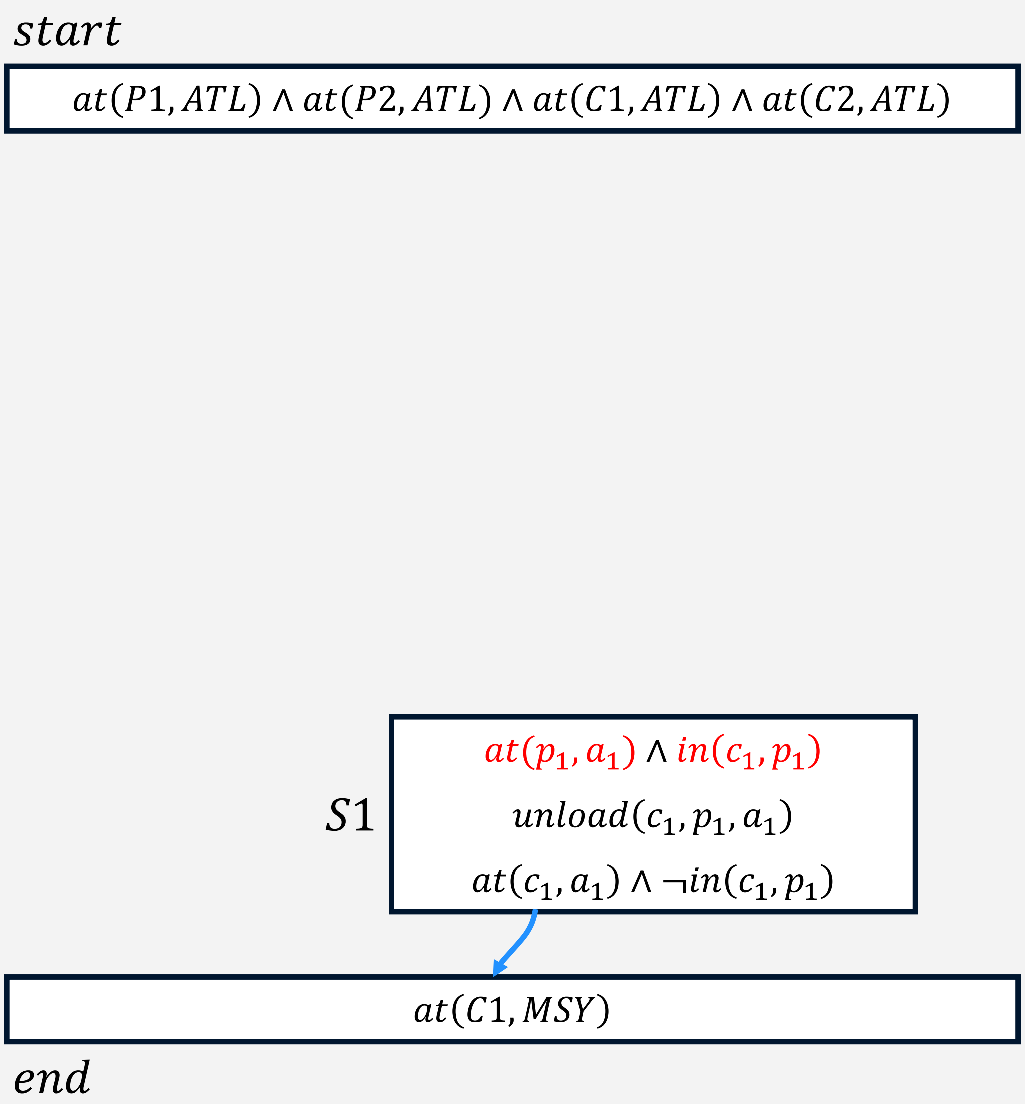
Flaw: $at(C1, MSY)$ open

Steps: $start$, end, $S1$

Bindings:

Orderings: $start < end$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$



$start$

$at(P1,ATL) \land at(P2,ATL) \land at(C1,ATL) \land at(C2,ATL)$

$S1$

$at(p_1,a_1) \land in(c_1,p_1)$

$unload(c_1,p_1,a_1)$

$at(c_1,a_1) \land \neg in(c_1,p_1)$

$at(C1,MSY)$

end

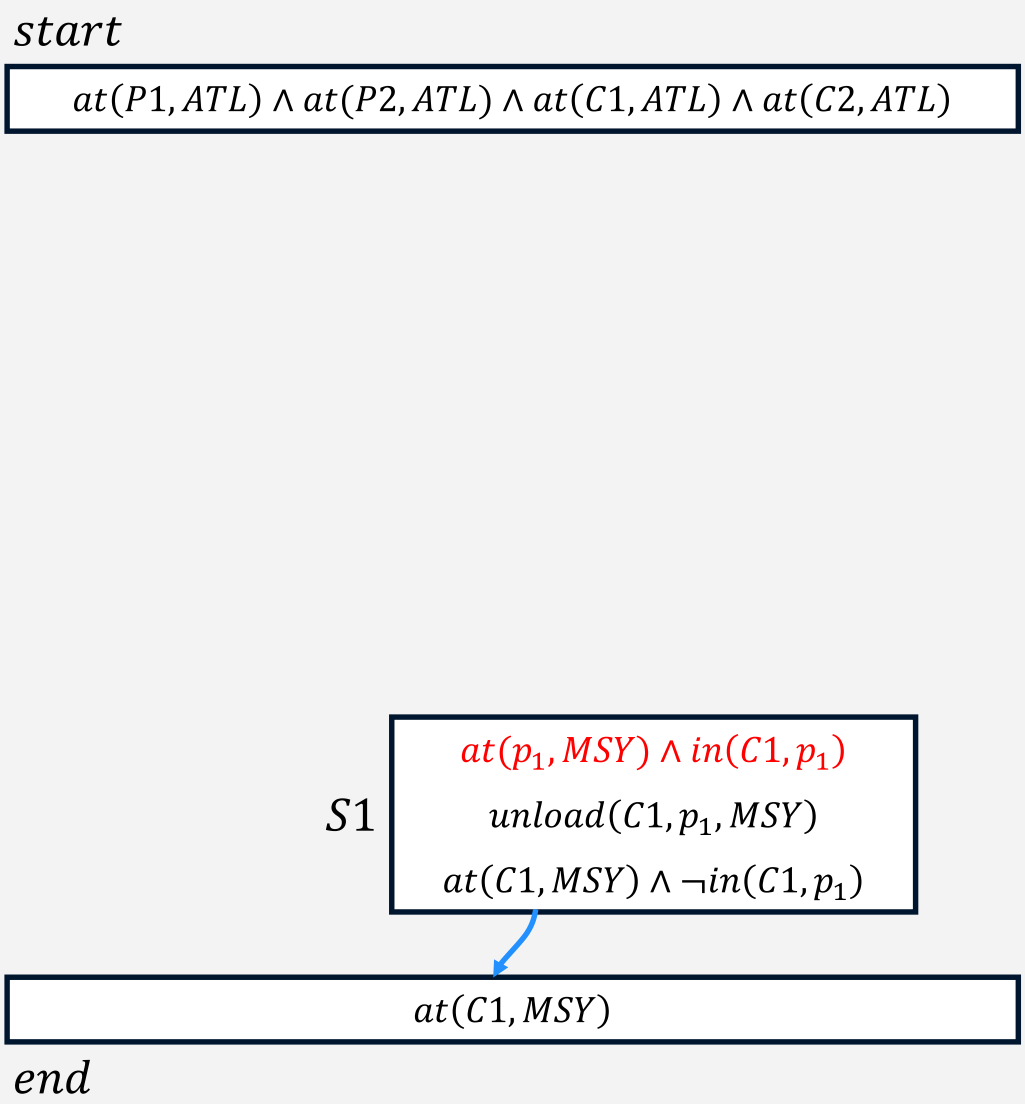
Flaw: $at(C1,MSY)$ open

Steps: $start$, end, $S1$

Bindings:

Orderings: $start < end$, $start < S1$, $S1 < end$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$



$start$

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$S1$

$at(p_1, MSY) \land in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \land \neg in(C1, p_1)$

$at(C1, MSY)$

end

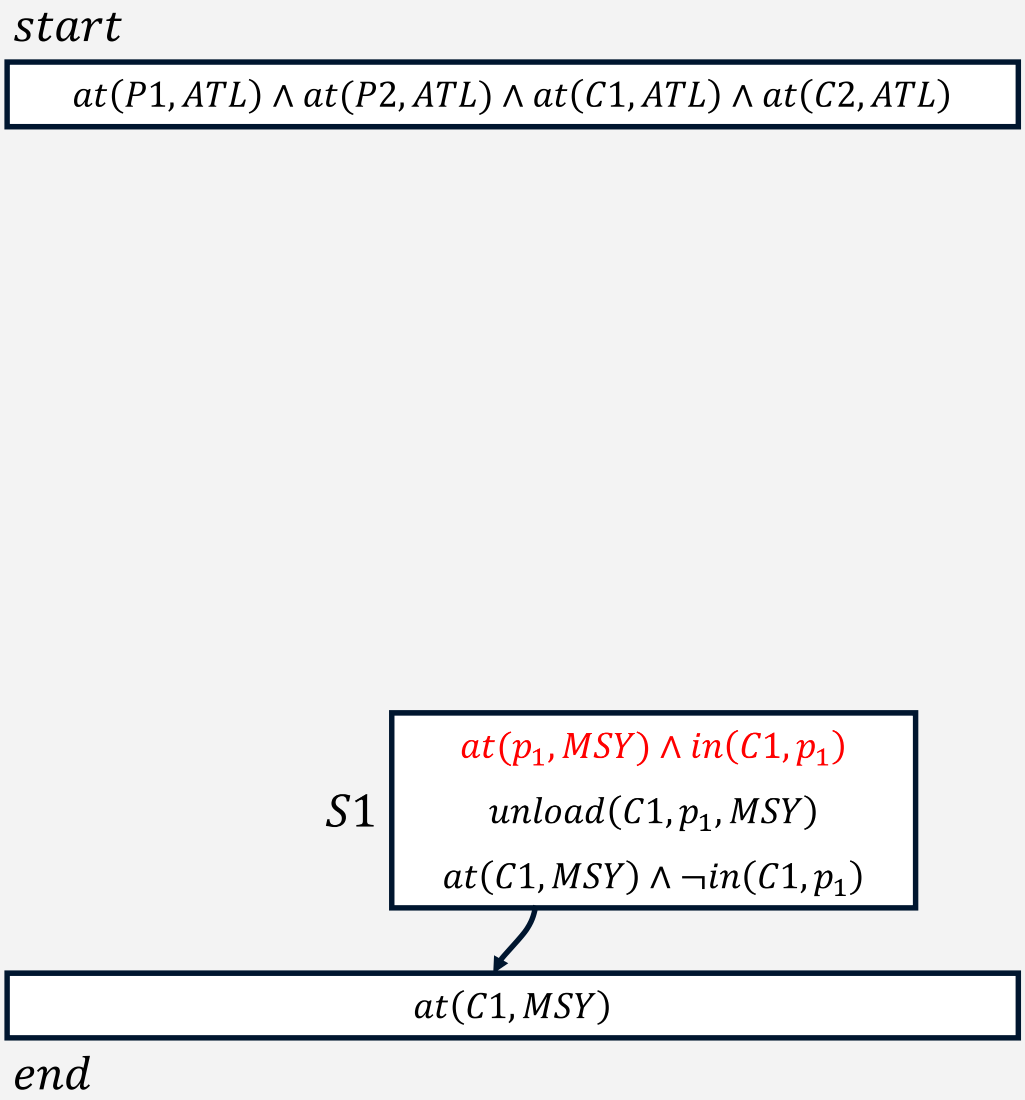
Flaw: $at(C1, MSY)$ open

Steps: $start$, end, $S1$

Bindings: $c_1 = C1$, $a_1 = MSY$

Orderings: $start < end$,
$start < S1$, $S1 < end$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$



$start$

$at(P1,ATL) \wedge at(P2,ATL) \wedge at(C1,ATL) \wedge at(C2,ATL)$

$S1$

$at(p_1,MSY) \wedge in(C1,p_1)$

$unload(C1,p_1,MSY)$

$at(C1,MSY) \wedge \neg in(C1,p_1)$

$at(C1,MSY)$

end

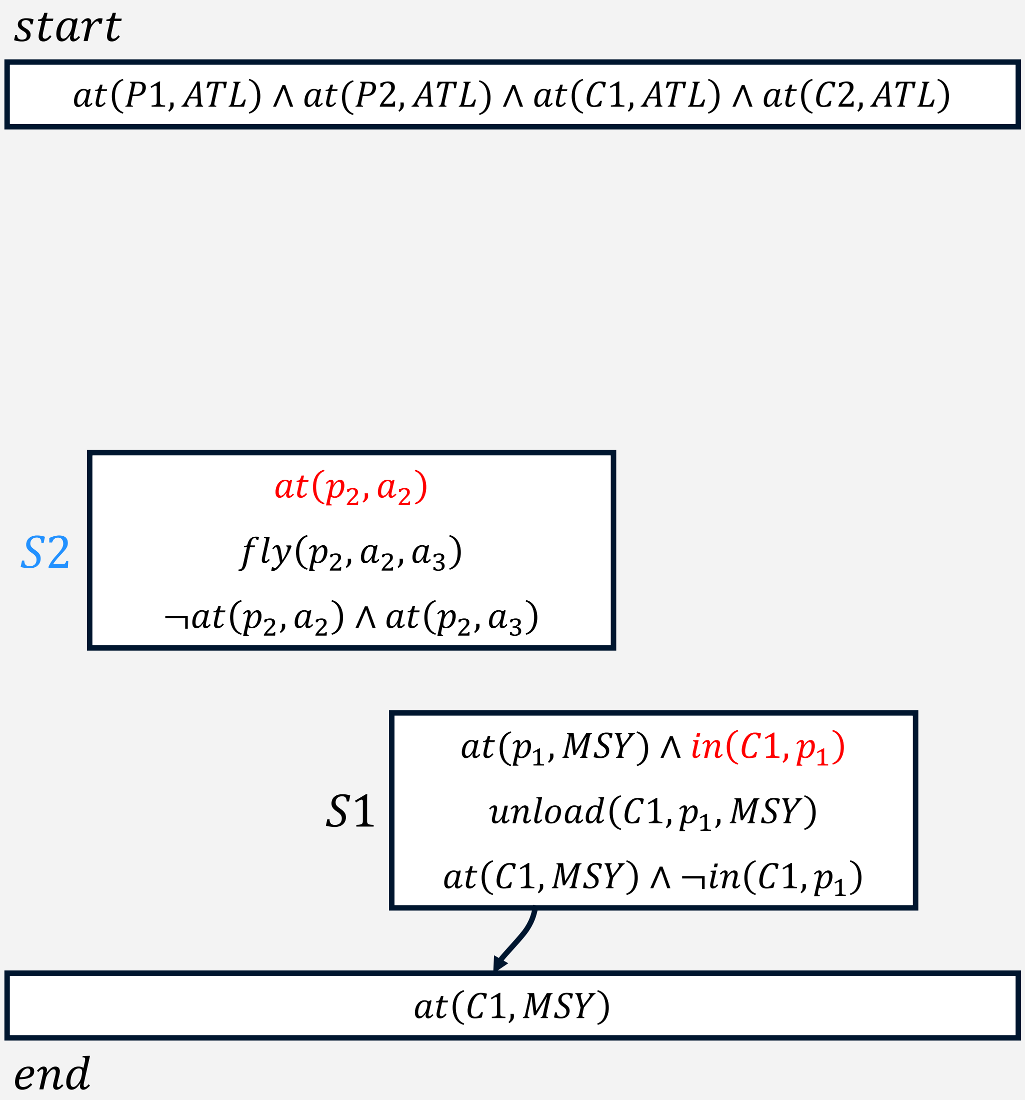
Flaw: $at(p_1,MSY)$ open

Steps: $start$, end, $S1$

Bindings: $c_1 = C1$, $a_1 = MSY$

Orderings: $start < end$, $start < S1$, $S1 < end$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$



start

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

*S*2

$at(p_2, a_2)$

$fly(p_2, a_2, a_3)$

$\neg at(p_2, a_2) \land at(p_2, a_3)$

*S*1

$at(p_1, MSY) \land in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \land \neg in(C1, p_1)$

$at(C1, MSY)$

end

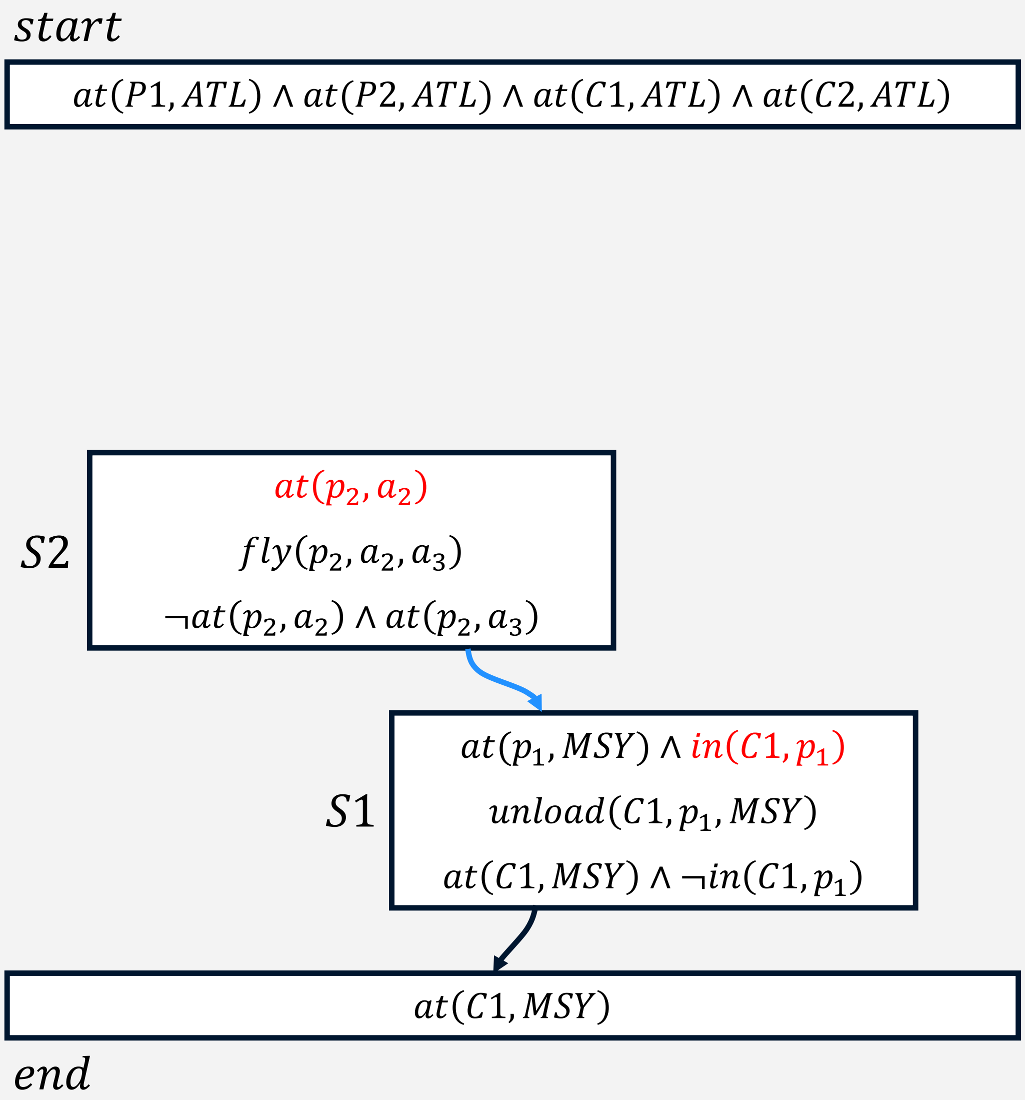
Flaw: $at(p_1, MSY)$ open

Steps: $start$, end, $S1$, $S2$

Bindings: $c_1 = C1$, $a_1 = MSY$

Orderings: $start < end$, $start < S1$, $S1 < end$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$



start

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

S2

$at(p_2, a_2)$

$fly(p_2, a_2, a_3)$

$\neg at(p_2, a_2) \land at(p_2, a_3)$

S1

$at(p_1, MSY) \land in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \land \neg in(C1, p_1)$

$at(C1, MSY)$

end

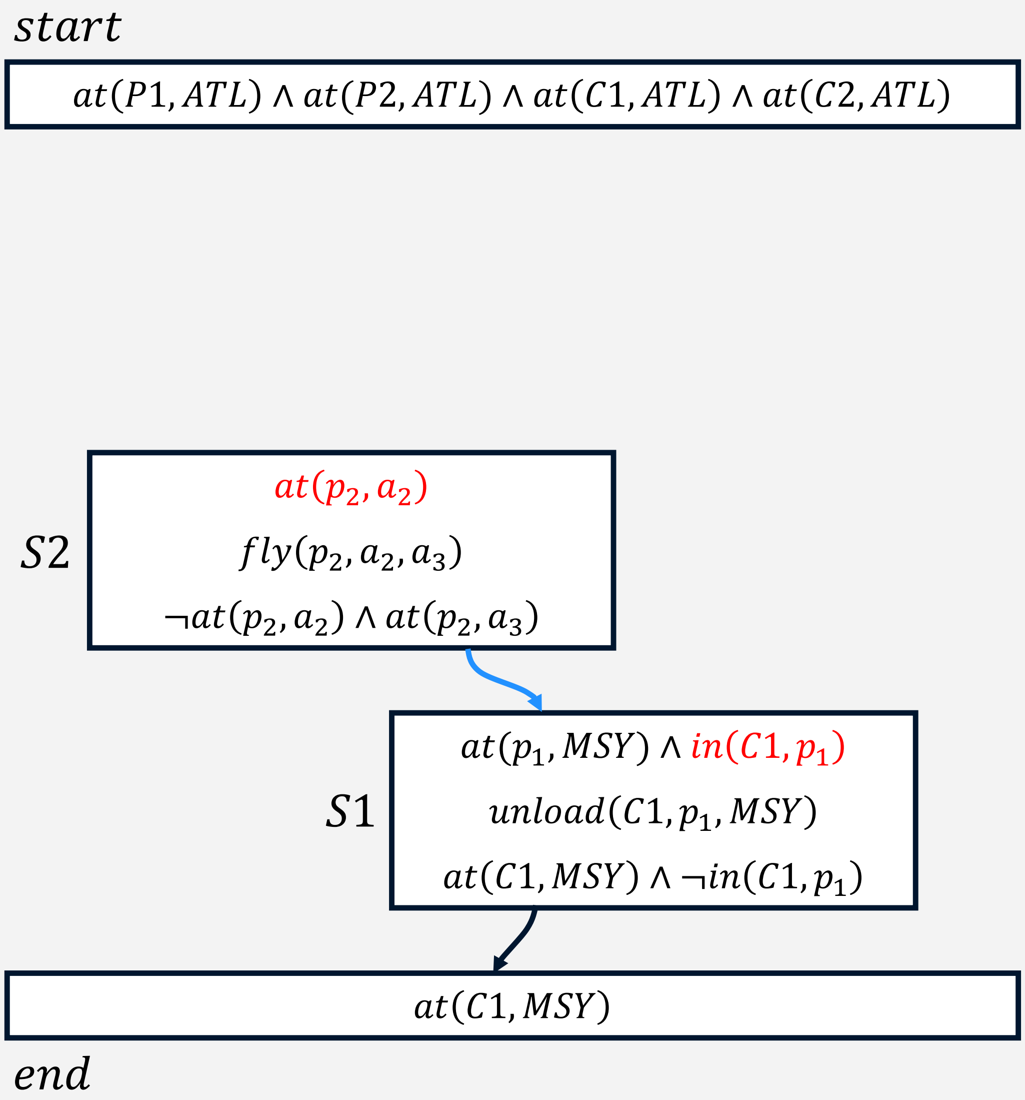
Flaw: $at(p_1, MSY)$ open

Steps: $start$, end, $S1$, $S2$

Bindings: $c_1 = C1$, $a_1 = MSY$

Orderings: $start < end$, $start < S1$, $S1 < end$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$



$start$

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$S2$

$at(p_2, a_2)$

$fly(p_2, a_2, a_3)$

$\neg at(p_2, a_2) \land at(p_2, a_3)$

$S1$

$at(p_1, MSY) \land in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \land \neg in(C1, p_1)$

$at(C1, MSY)$

end

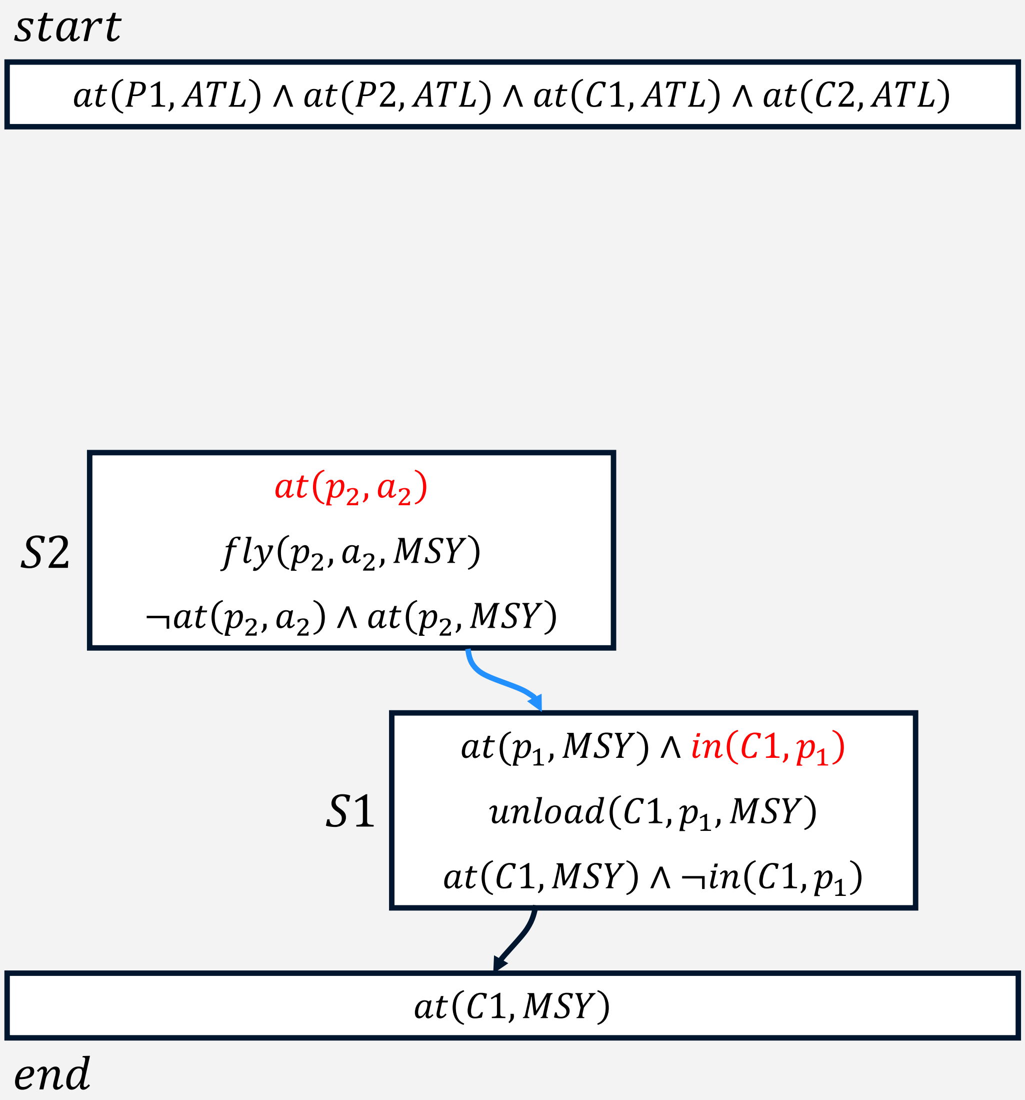
Flaw: $at(p_1, MSY)$ open

Steps: $start$, end, $S1$, $S2$

Bindings: $c_1 = C1$, $a_1 = MSY$

Orderings: $start < end$,
$start < S1$, $S1 < end$,
$start < S2$, $S2 < end$, $S2 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$,
$S2 \xrightarrow{at(P1,MSY)} S1$



$start$

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$S2$

$at(p_2, a_2)$

$fly(p_2, a_2, MSY)$

$\neg at(p_2, a_2) \land at(p_2, MSY)$

$S1$

$at(p_1, MSY) \land in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \land \neg in(C1, p_1)$

$at(C1, MSY)$

end

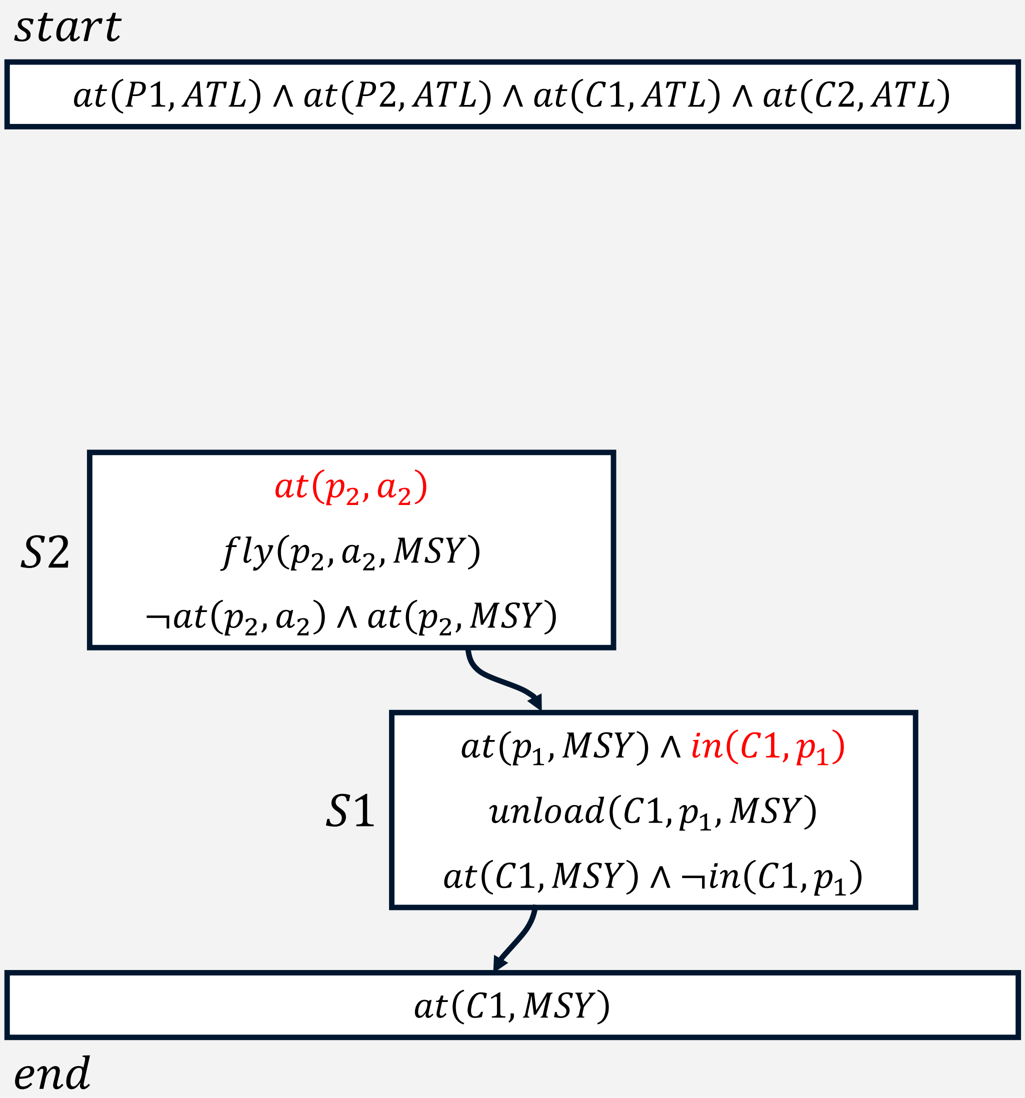
Flaw: $at(p_1, MSY)$ open

Steps: $start$, end, $S1$, $S2$

Bindings: $c_1 = C1$, $a_1 = MSY$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$



$start$

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$S2$

$at(p_2, a_2)$

$fly(p_2, a_2, MSY)$

$\neg at(p_2, a_2) \wedge at(p_2, MSY)$

$S1$

$at(p_1, MSY) \wedge in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, p_1)$

$at(C1, MSY)$

end

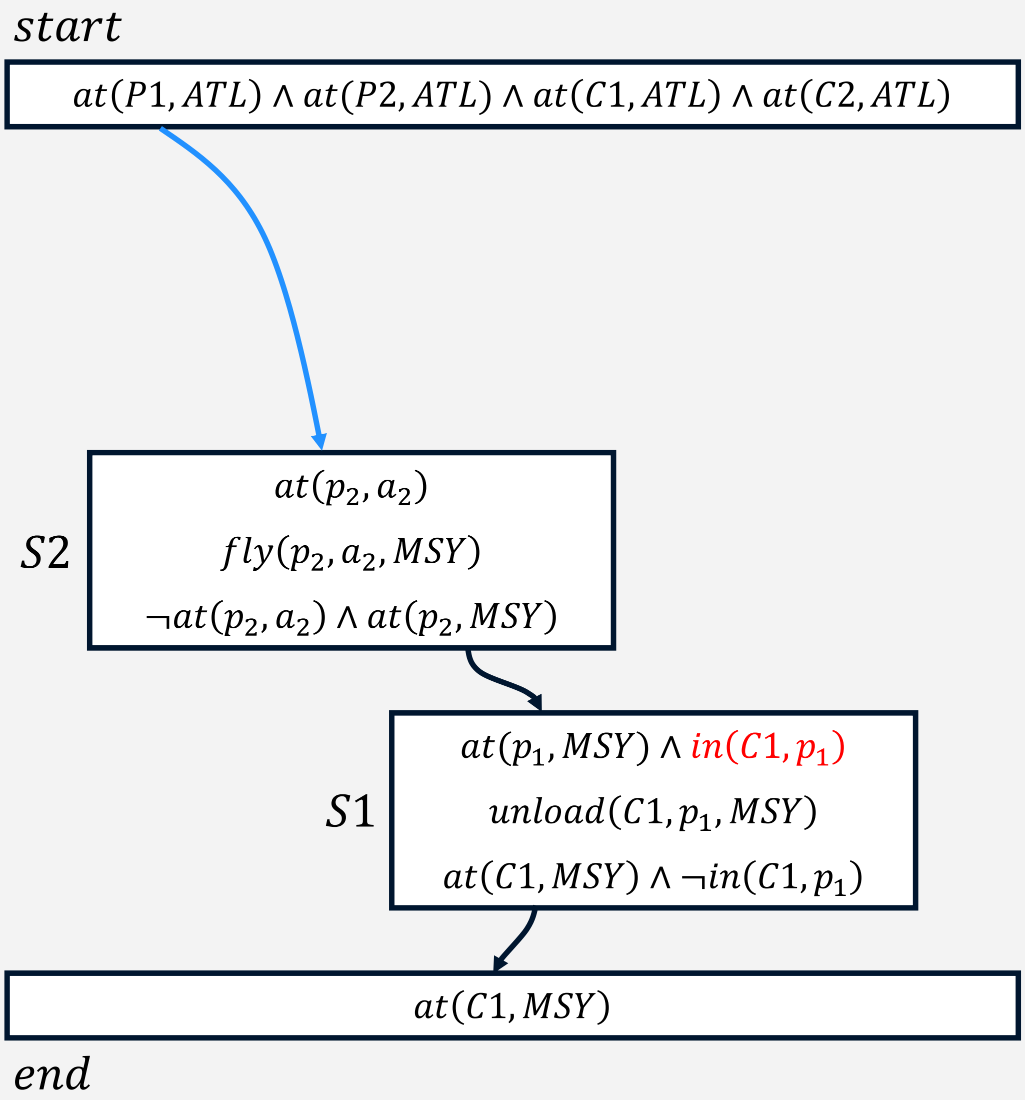
Flaw: $at(p_2, a_2)$ open

Steps: $start$, end, $S1$, $S2$

Bindings: $c_1 = C1$, $a_1 = MSY$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$



start

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

S2

$at(p_2, a_2)$

$fly(p_2, a_2, MSY)$

$\neg at(p_2, a_2) \land at(p_2, MSY)$

S1

$at(p_1, MSY) \land in(C1, p_1)$

$unload(C1, p_1, MSY)$

$at(C1, MSY) \land \neg in(C1, p_1)$

$at(C1, MSY)$

end

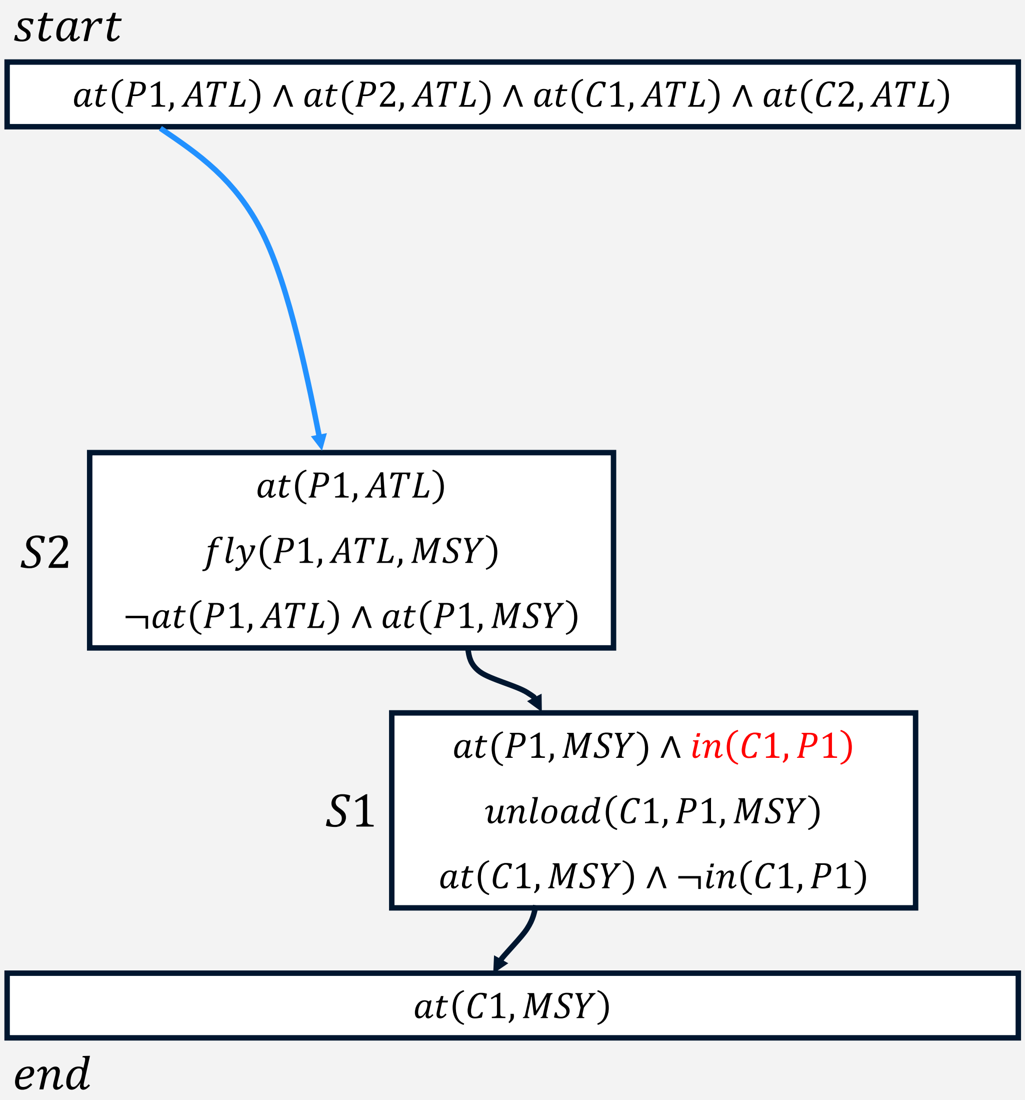
Flaw: $at(p_2, a_2)$ open

Steps: $start$, end, $S1$, $S2$

Bindings: $c_1 = C1$, $a_1 = MSY$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$



start

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

S2

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

S1

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$

end

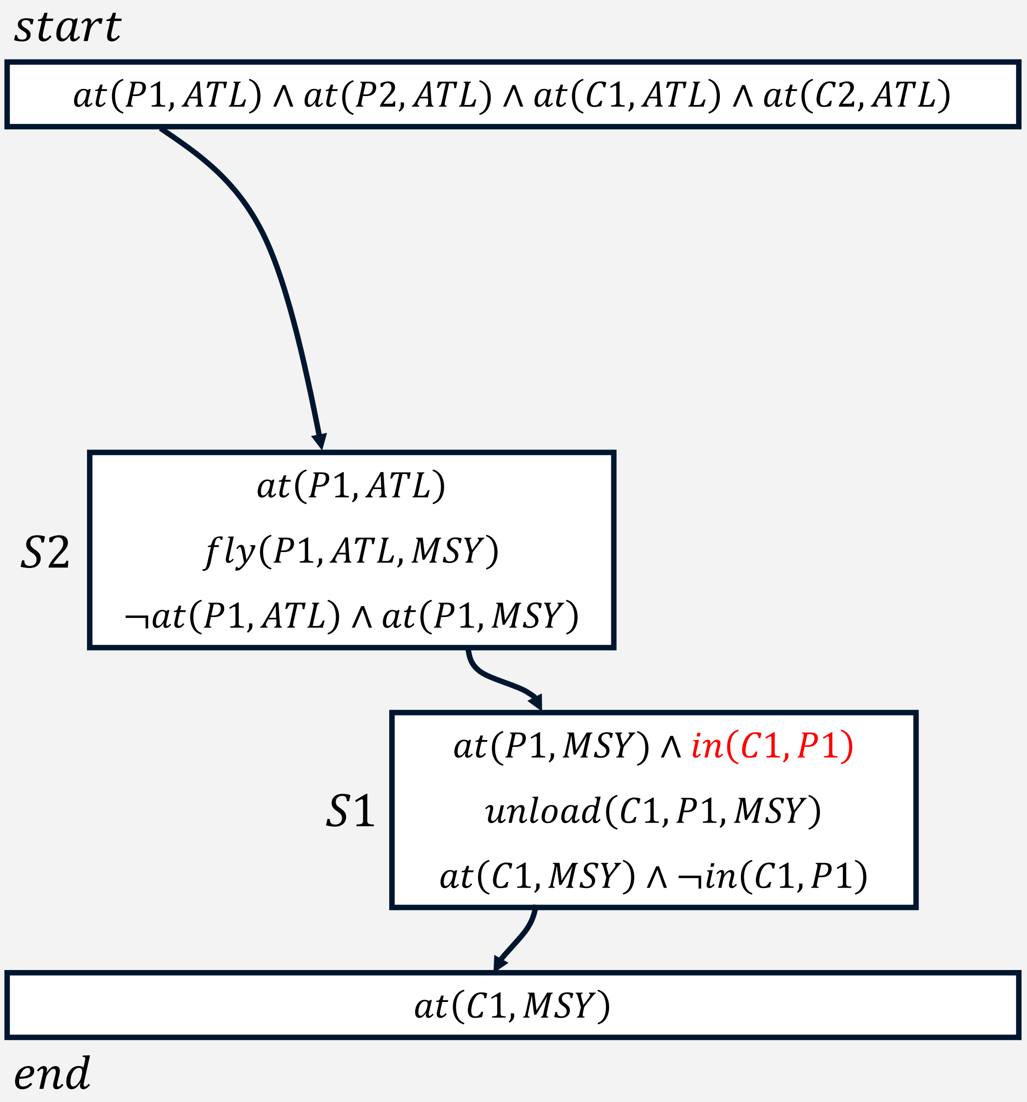
Flaw: $at(p_2, a_2)$ open

Steps: $start$, end, $S1$, $S2$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$



start

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \land at(P1, MSY)$

$S1$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

$at(C1, MSY)$

end

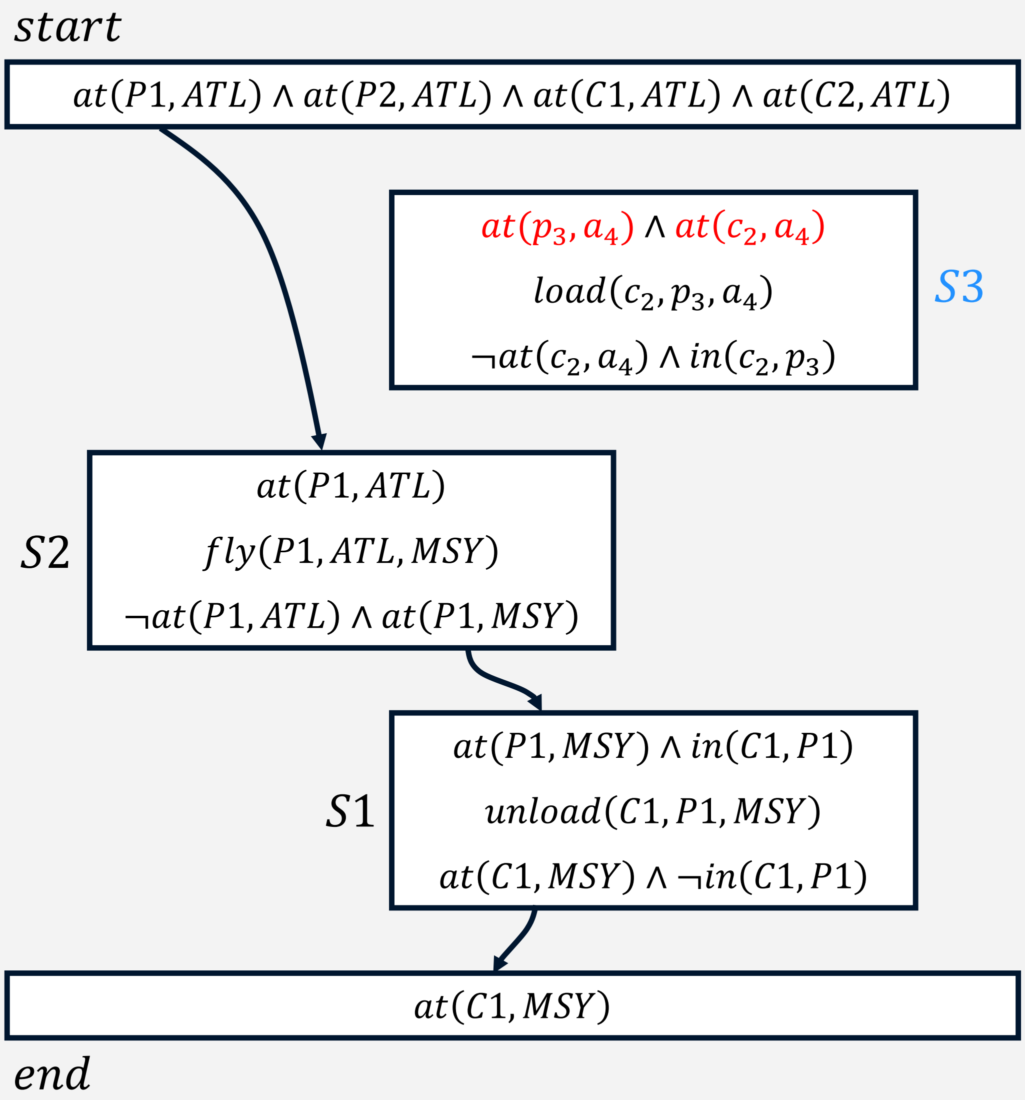
Flaw: $in(C1, P1)$ open

Steps: $start$, end, $S1$, $S2$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$



start

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(p_3, a_4) \land at(c_2, a_4)$

$load(c_2, p_3, a_4)$

$\neg at(c_2, a_4) \land in(c_2, p_3)$

$S3$

$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \land at(P1, MSY)$

$S1$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

$at(C1, MSY)$

end

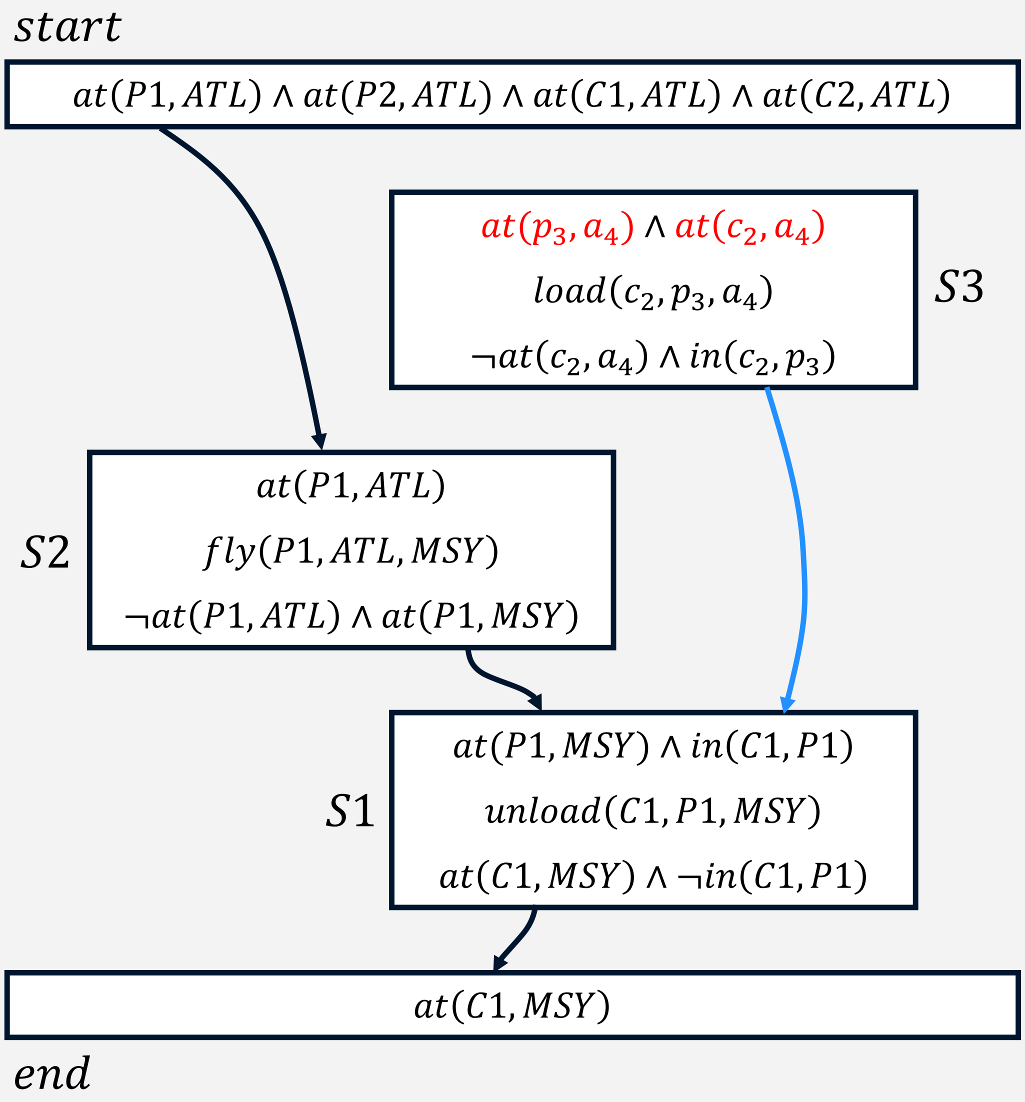
Flaw: $in(C1, P1)$ open

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$



$start$

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$S3$

$at(p_3, a_4) \land at(c_2, a_4)$

$load(c_2, p_3, a_4)$

$\neg at(c_2, a_4) \land in(c_2, p_3)$

$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \land at(P1, MSY)$

$S1$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

$at(C1, MSY)$

end

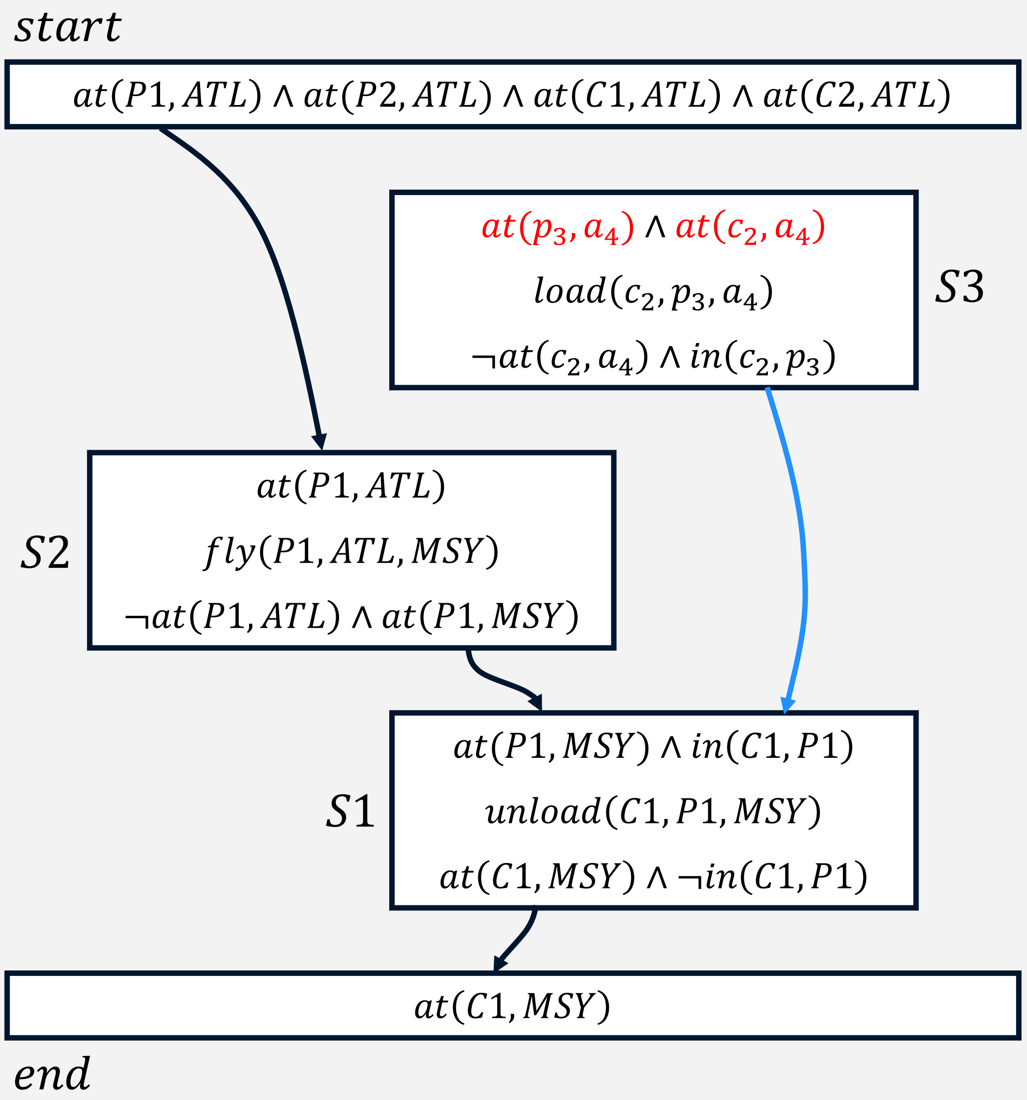
Flaw: $in(C1, P1)$ open

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$, $S3 \xrightarrow{in(C1,P1)} S1$


$start$
$at(P1,ATL) \land at(P2,ATL) \land at(C1,ATL) \land at(C2,ATL)$
$at(p_3,a_4) \land at(c_2,a_4)$
$load(c_2,p_3,a_4)$
$\neg at(c_2,a_4) \land in(c_2,p_3)$
$S3$
$at(P1,ATL)$
$fly(P1,ATL,MSY)$
$\neg at(P1,ATL) \land at(P1,MSY)$
$S2$
$at(P1,MSY) \land in(C1,P1)$
$unload(C1,P1,MSY)$
$at(C1,MSY) \land \neg in(C1,P1)$
$S1$
$at(C1,MSY)$
end

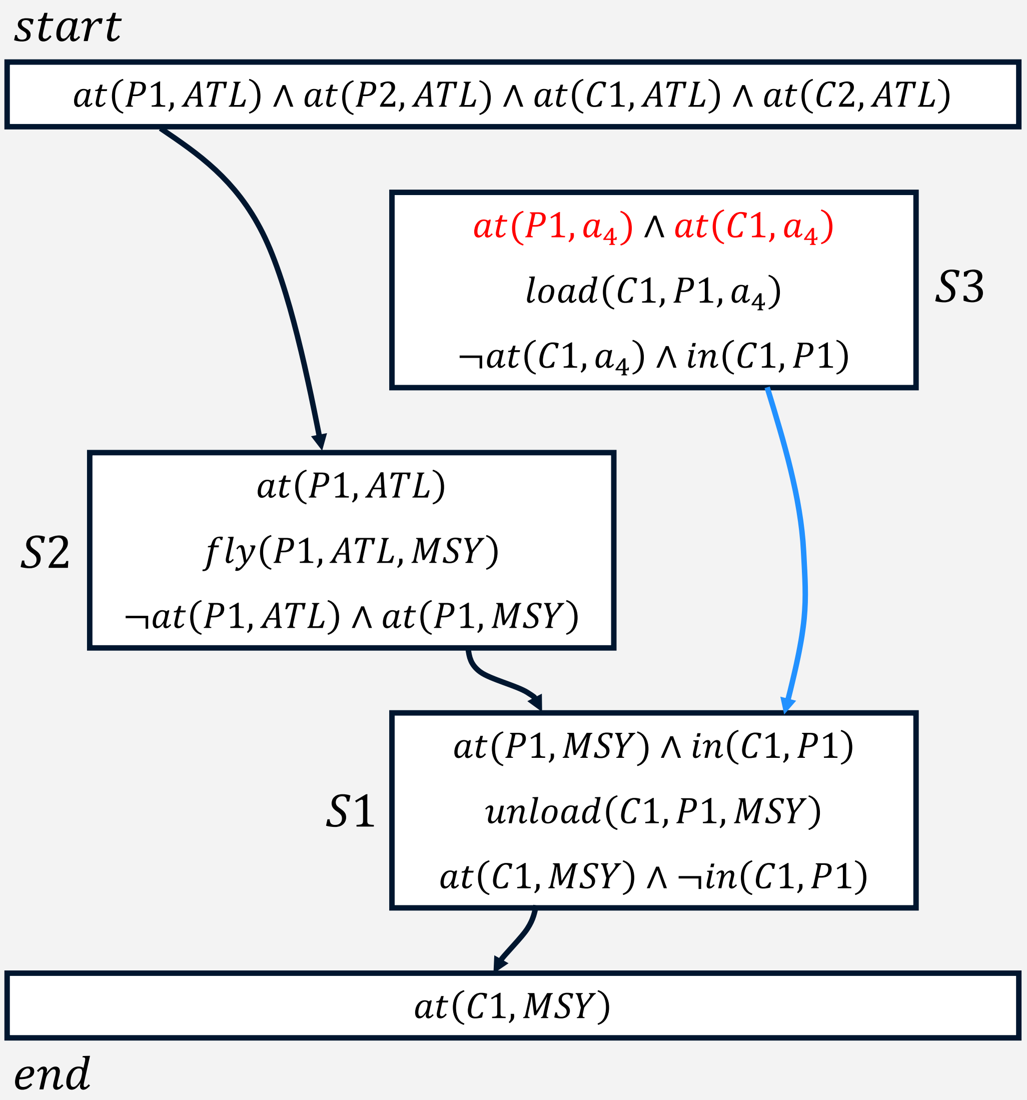
Flaw: $in(C1,P1)$ open

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$,
$a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$,
$a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$

Orderings: $start < end$,
$start < S1$, $S1 < end$,
$start < S2$, $S2 < end$, $S2 < S1$,
$start < S3$, $S3 < end$, $S3 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$,
$S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$,
$S3 \xrightarrow{in(C1,P1)} S1$



$start$

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$at(P1, a_4) \land at(C1, a_4)$

$load(C1, P1, a_4)$

$\neg at(C1, a_4) \land in(C1, P1)$

$S3$

$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \land at(P1, MSY)$

$S1$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

$at(C1, MSY)$

end

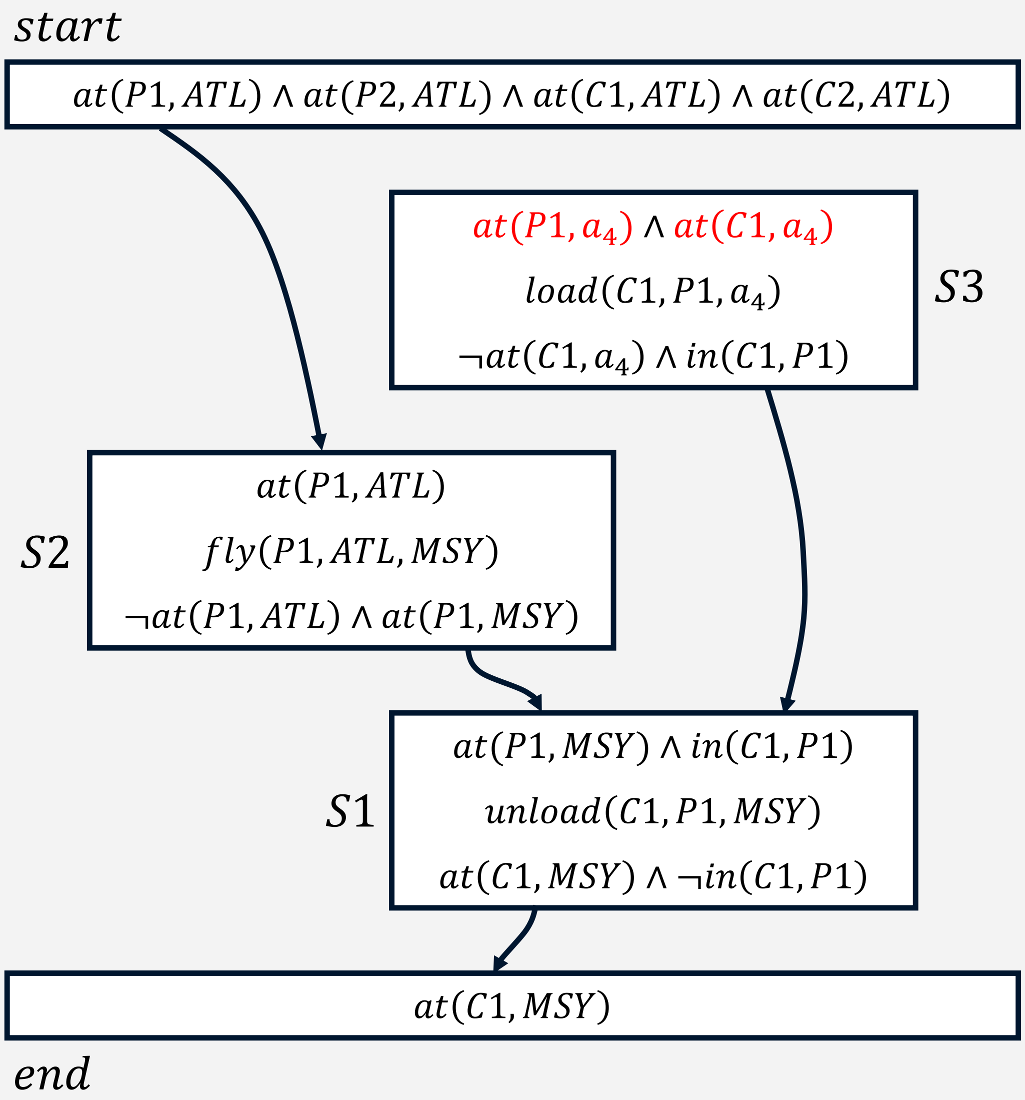
Flaw: $in(C1, P1)$ open

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1, p_1 = P1, a_1 = MSY, p_2 = P1, a_2 = ATL, a_3 = MSY, p_1 = p_2, a_1 = a_3, c_2 = C1, p_3 = P1, c_1 = c_2, p_1 = p_3, a_1 = a_4$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$, $start < S3$, $S3 < end$, $S3 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$, $S3 \xrightarrow{in(C1,P1)} S1$



$start$

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$S3$

$at(P1, a_4) \wedge at(C1, a_4)$

$load(C1, P1, a_4)$

$\neg at(C1, a_4) \wedge in(C1, P1)$

$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

$S1$

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$

end

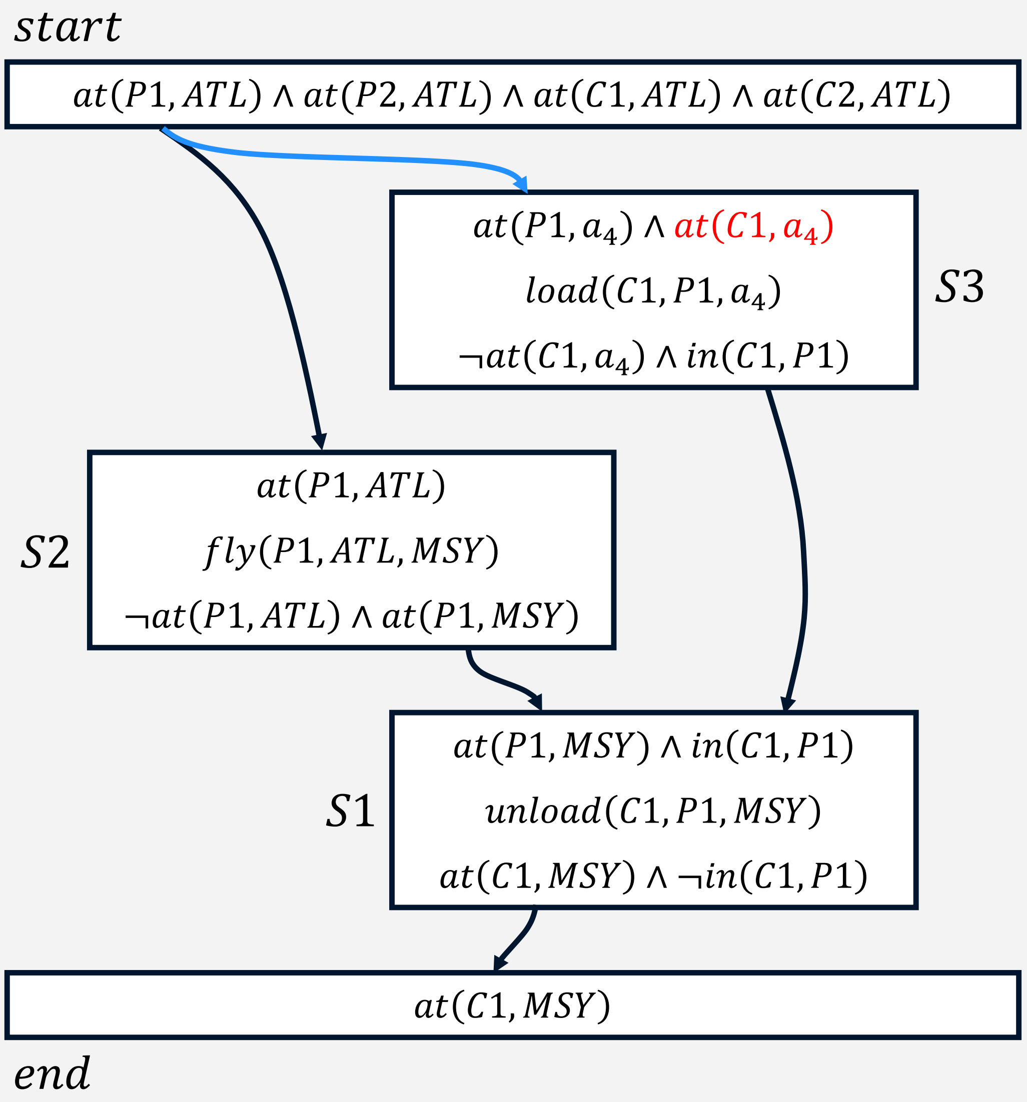
Flaw: $at(P1, a_4)$ open

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$, $c_2 = C1$, $p_3 = P1$, $c_1 = c_2$, $p_1 = p_3$, $a_1 = a_4$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$, $start < S3$, $S3 < end$, $S3 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$, $S3 \xrightarrow{in(C1,P1)} S1$


$start$
$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$
$at(P1, a_4) \land at(C1, a_4)$
$load(C1, P1, a_4)$
$\neg at(C1, a_4) \land in(C1, P1)$
$S3$
$S2$
$at(P1, ATL)$
$fly(P1, ATL, MSY)$
$\neg at(P1, ATL) \land at(P1, MSY)$
$S1$
$at(P1, MSY) \land in(C1, P1)$
$unload(C1, P1, MSY)$
$at(C1, MSY) \land \neg in(C1, P1)$
$at(C1, MSY)$
end

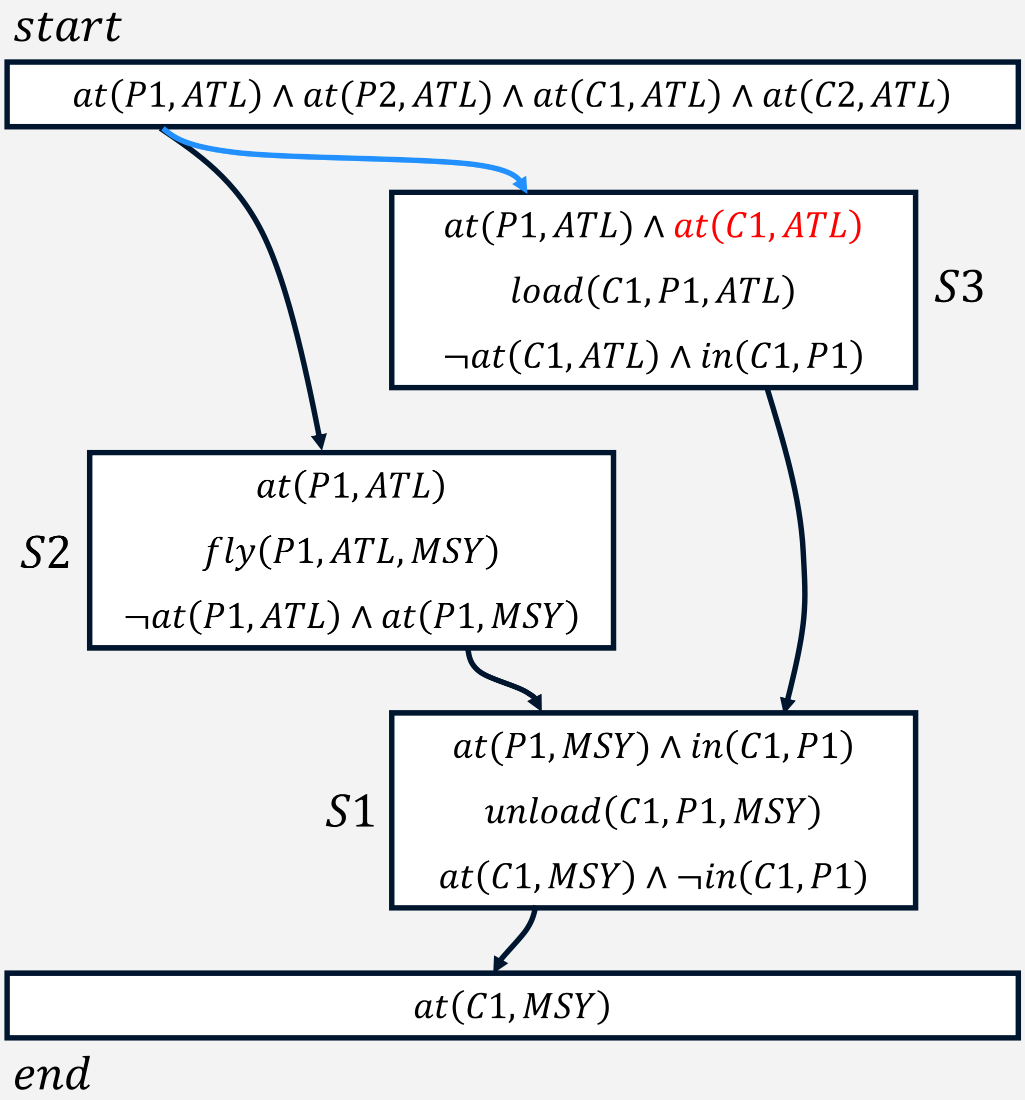
Flaw: $at(P1, a_4)$ open

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$, $c_2 = C1$, $p_3 = P1$, $c_1 = c_2$, $p_1 = p_3$, $a_1 = a_4$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$, $start < S3$, $S3 < end$, $S3 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$, $S3 \xrightarrow{in(C1,P1)} S1$, $start \xrightarrow{at(P1,ATL)} S3$



start

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

S3

$at(P1, ATL) \wedge at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \wedge in(C1, P1)$

S2

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

S1

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$

end

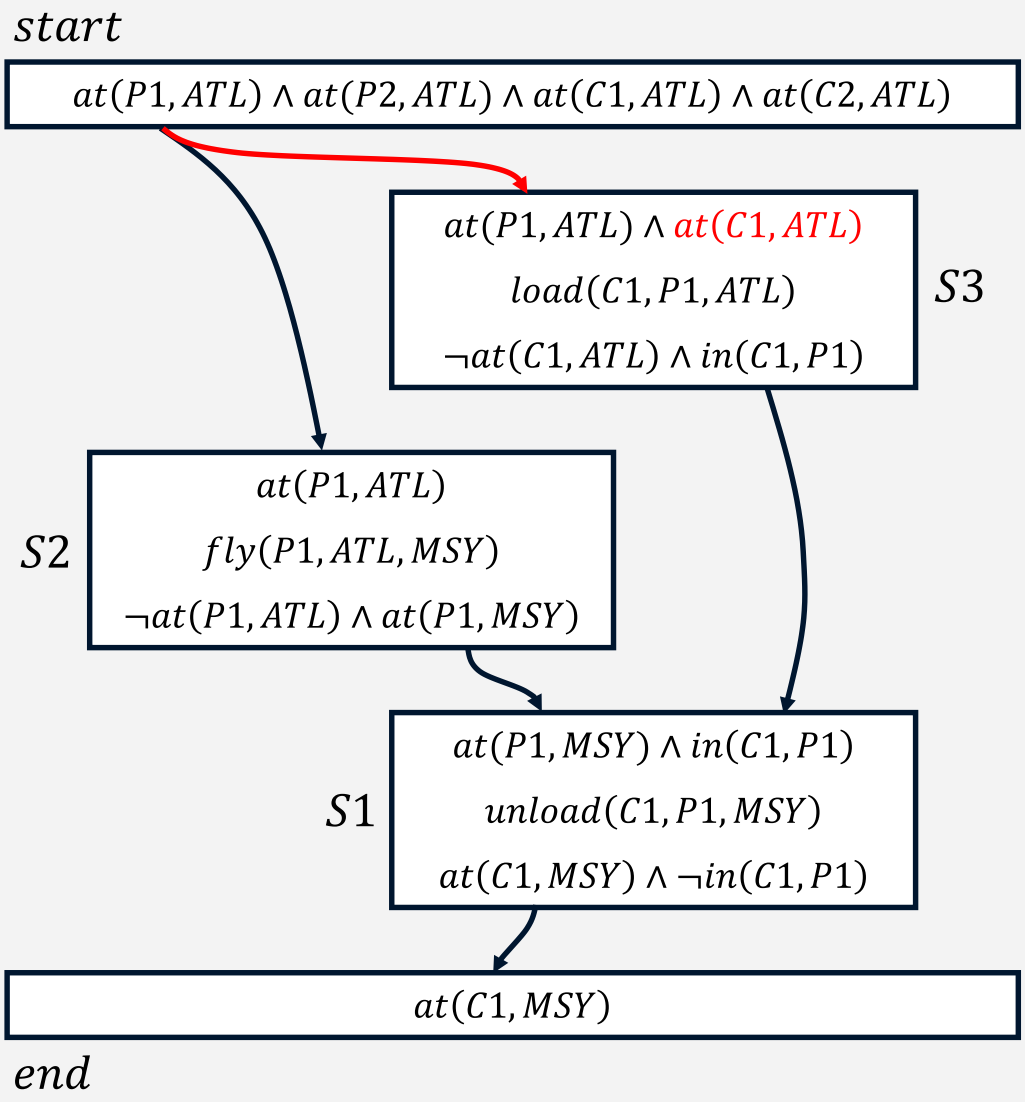
Flaw: $at(P1, a_4)$ open

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$, $c_2 = C1$, $p_3 = P1$, $a_4 = ATL$, $c_1 = c_2$, $p_1 = p_3$, $a_1 = a_4$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$, $start < S3$, $S3 < end$, $S3 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$, $S3 \xrightarrow{in(C1,P1)} S1$, $start \xrightarrow{at(P1,ATL)} S3$



$start$

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$S3$

$at(P1, ATL) \land at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \land in(C1, P1)$

$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \land at(P1, MSY)$

$S1$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

$at(C1, MSY)$

end

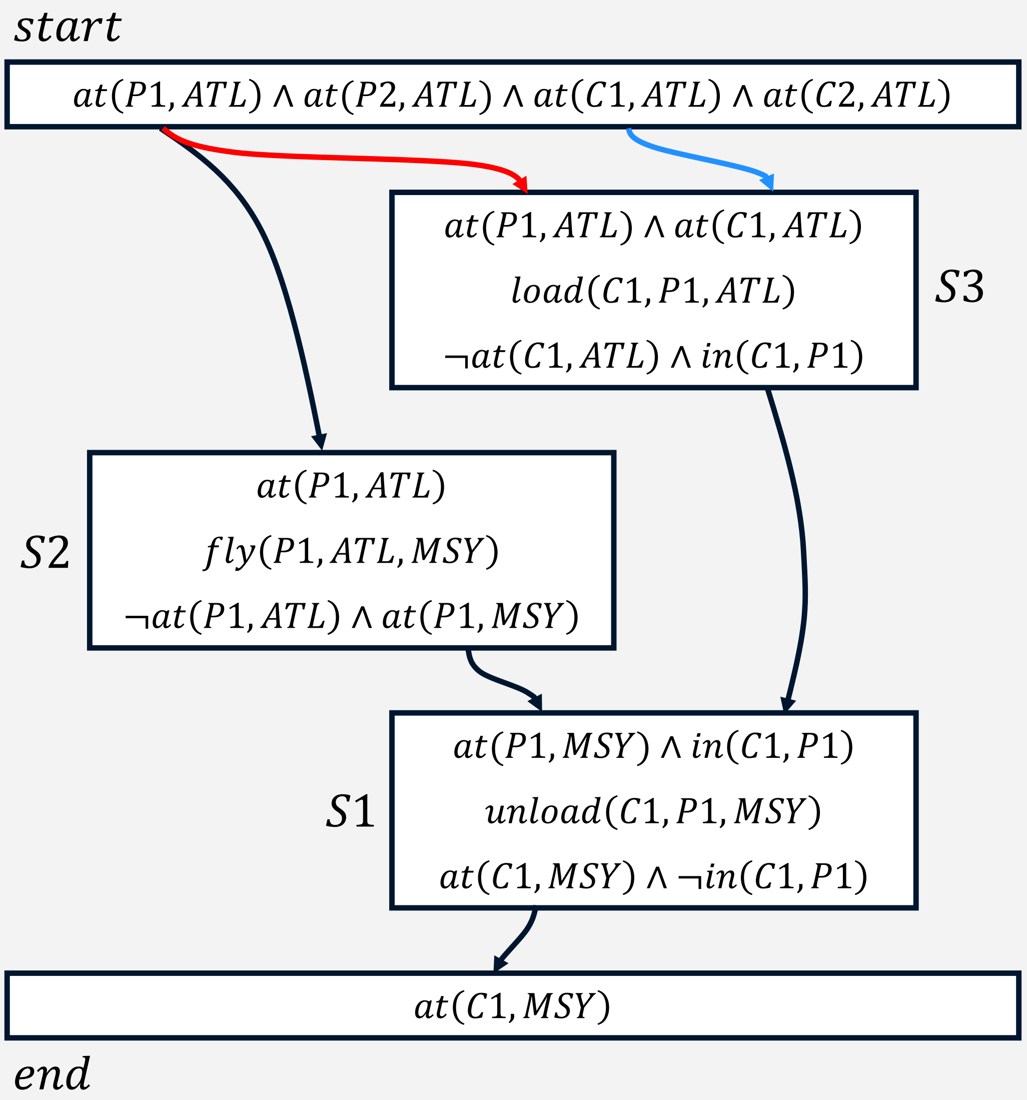
Flaw: $at(C1, ATL)$ open

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$,
$a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$,
$a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$,
$c_2 = C1$, $p_3 = P1$, $a_4 = ATL$,
$c_1 = c_2$, $p_1 = p_3$, $a_1 = a_4$

Orderings: $start < end$,
$start < S1$, $S1 < end$,
$start < S2$, $S2 < end$, $S2 < S1$,
$start < S3$, $S3 < end$, $S3 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$,
$S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$,
$S3 \xrightarrow{in(C1,P1)} S1$, $start \xrightarrow{at(P1,ATL)} S3$



$start$

$at(P1, ATL) \land at(P2, ATL) \land at(C1, ATL) \land at(C2, ATL)$

$S3$

$at(P1, ATL) \land at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \land in(C1, P1)$

$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \land at(P1, MSY)$

$S1$

$at(P1, MSY) \land in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \land \neg in(C1, P1)$

$at(C1, MSY)$

end

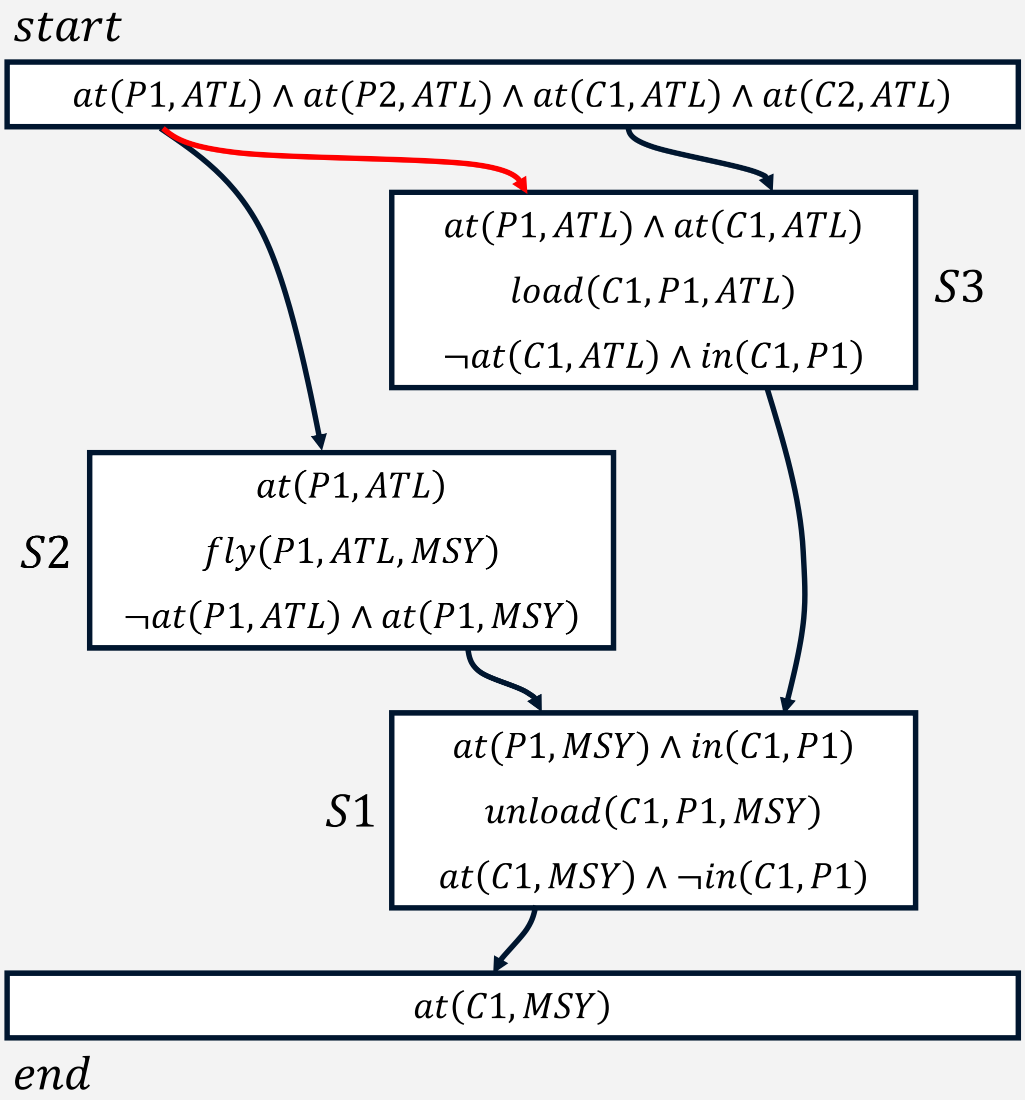
Flaw: $at(C1, ATL)$ open

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$, $c_2 = C1$, $p_3 = P1$, $a_4 = ATL$, $c_1 = c_2$, $p_1 = p_3$, $a_1 = a_4$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$, $start < S3$, $S3 < end$, $S3 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$, $S3 \xrightarrow{in(C1,P1)} S1$, $start \xrightarrow{at(P1,ATL)} S3$, $start \xrightarrow{at(C1,ATL)} S3$


$start$
$at(P1,ATL) \land at(P2,ATL) \land at(C1,ATL) \land at(C2,ATL)$
$at(P1,ATL) \land at(C1,ATL)$
$load(C1,P1,ATL)$
$\neg at(C1,ATL) \land in(C1,P1)$
$S3$
$S2$
$at(P1,ATL)$
$fly(P1,ATL,MSY)$
$\neg at(P1,ATL) \land at(P1,MSY)$
$S1$
$at(P1,MSY) \land in(C1,P1)$
$unload(C1,P1,MSY)$
$at(C1,MSY) \land \neg in(C1,P1)$
$at(C1,MSY)$
end

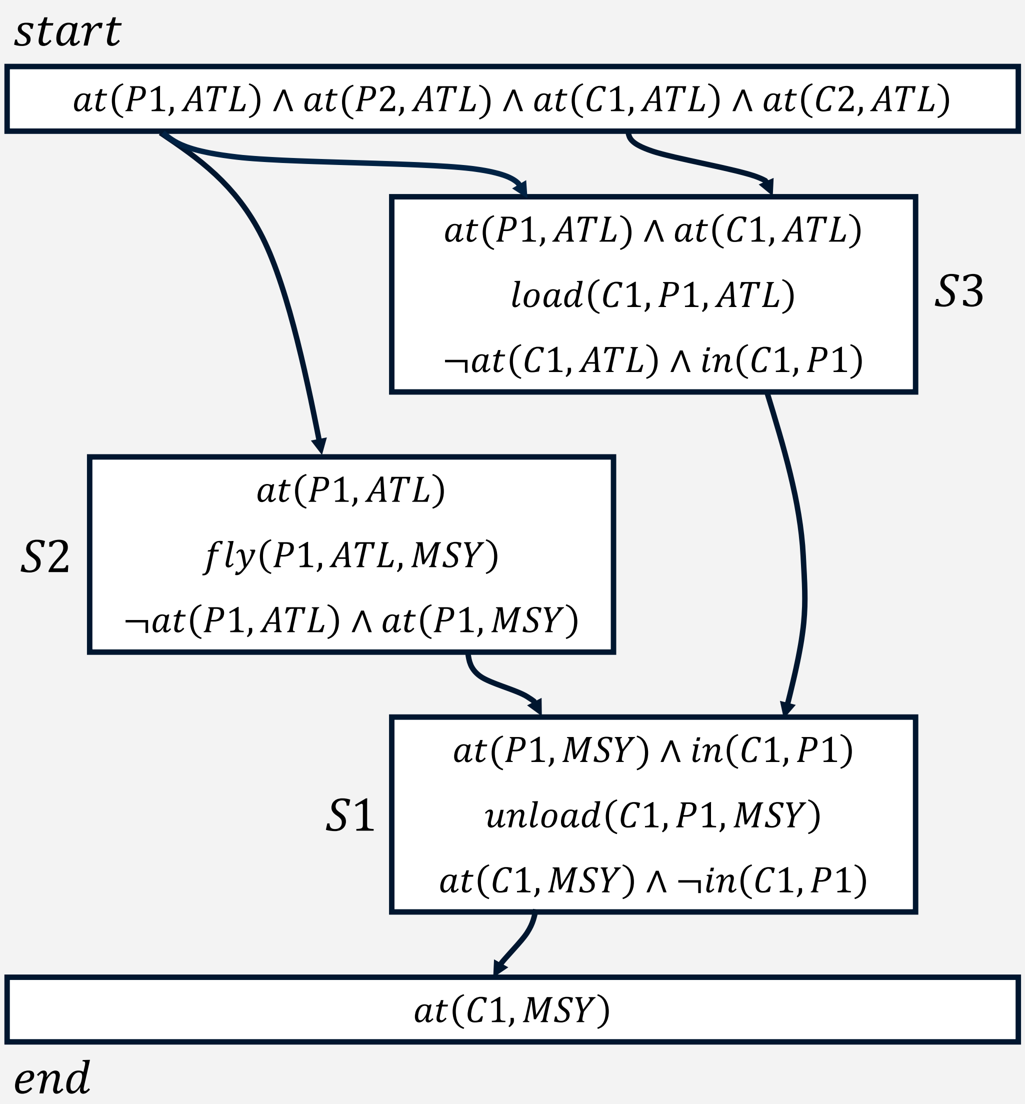
Flaw: $S2$ threatens $start \rightarrow S3$

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$, $c_2 = C1$, $p_3 = P1$, $a_4 = ATL$, $c_1 = c_2$, $p_1 = p_3$, $a_1 = a_4$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$, $start < S3$, $S3 < end$, $S3 < S1$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$, $S3 \xrightarrow{in(C1,P1)} S1$, $start \xrightarrow{at(P1,ATL)} S3$, $start \xrightarrow{at(C1,ATL)} S3$



$start$

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$S3$

$at(P1, ATL) \wedge at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \wedge in(C1, P1)$

$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

$S1$

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$

end

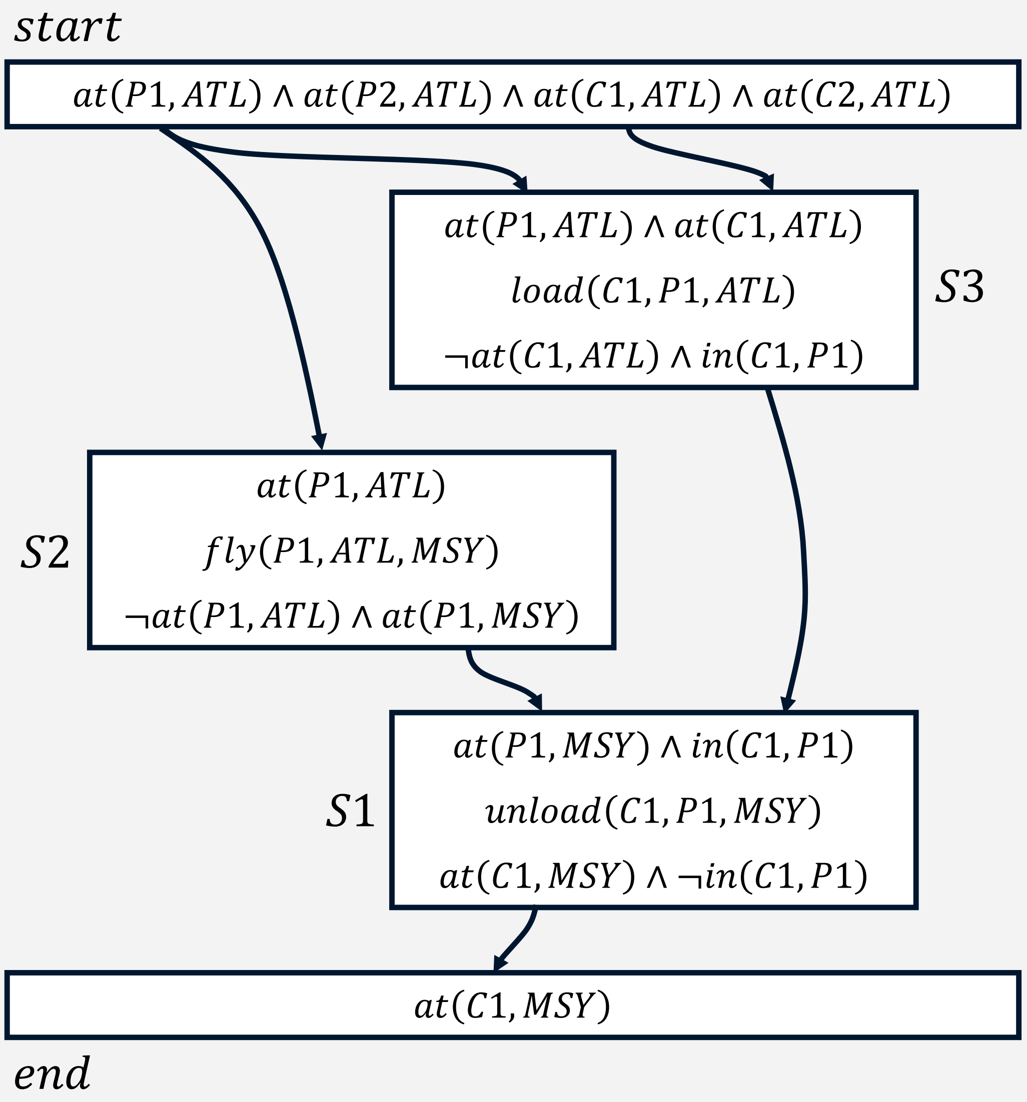
Flaw: $S2$ threatens $start \rightarrow S3$

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$, $c_2 = C1$, $p_3 = P1$, $a_4 = ATL$, $c_1 = c_2$, $p_1 = p_3$, $a_1 = a_4$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$, $start < S3$, $S3 < end$, $S3 < S1$, $S3 < S2$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$, $S3 \xrightarrow{in(C1,P1)} S1$, $start \xrightarrow{at(P1,ATL)} S3$, $start \xrightarrow{at(C1,ATL)} S3$



$start$

$at(P1, ATL) \wedge at(P2, ATL) \wedge at(C1, ATL) \wedge at(C2, ATL)$

$S3$

$at(P1, ATL) \wedge at(C1, ATL)$

$load(C1, P1, ATL)$

$\neg at(C1, ATL) \wedge in(C1, P1)$

$S2$

$at(P1, ATL)$

$fly(P1, ATL, MSY)$

$\neg at(P1, ATL) \wedge at(P1, MSY)$

$S1$

$at(P1, MSY) \wedge in(C1, P1)$

$unload(C1, P1, MSY)$

$at(C1, MSY) \wedge \neg in(C1, P1)$

$at(C1, MSY)$

end

Flaw: None!

Steps: $start$, end, $S1$, $S2$, $S3$

Bindings: $c_1 = C1$, $p_1 = P1$, $a_1 = MSY$, $p_2 = P1$, $a_2 = ATL$, $a_3 = MSY$, $p_1 = p_2$, $a_1 = a_3$, $c_2 = C1$, $p_3 = P1$, $a_4 = ATL$, $c_1 = c_2$, $p_1 = p_3$, $a_1 = a_4$

Orderings: $start < end$, $start < S1$, $S1 < end$, $start < S2$, $S2 < end$, $S2 < S1$, $start < S3$, $S3 < end$, $S3 < S1$, $S3 < S2$

Causal Links: $S1 \xrightarrow{at(C1,MSY)} end$, $S2 \xrightarrow{at(P1,MSY)} S1$, $start \xrightarrow{at(P1,ATL)} S2$, $S3 \xrightarrow{in(C1,P1)} S1$, $start \xrightarrow{at(P1,ATL)} S3$, $start \xrightarrow{at(C1,ATL)} S3$