

State-Space Planning

Stephen G. Ware

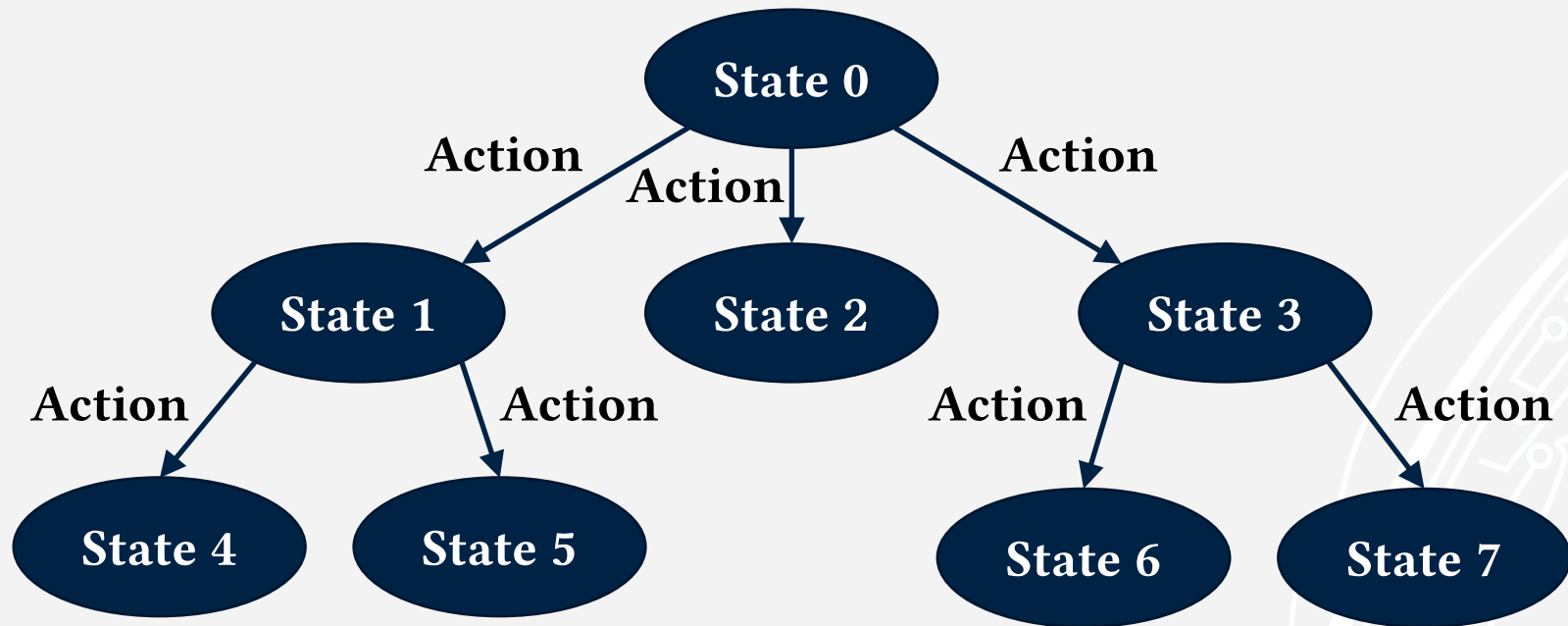
CS 463G



Short History of Planning

- State-space planning seems the most intuitive way to make planning into a search problem.
- However, a planning problem's state space explodes so quickly that we can't hope to search it unless we have a very accurate heuristic.
- We can reduce the number of decisions we have to make during planning through abstraction.
- Plan graphs provide an abstraction of a planning problem's search space.
- Plan graphs lead to accurate heuristics!

State-Space Search



State-Space Planner

Begin with an empty priority queue.

Put the initial state onto the priority queue.

While the queue is not empty:

- Pop a state C off the priority queue.

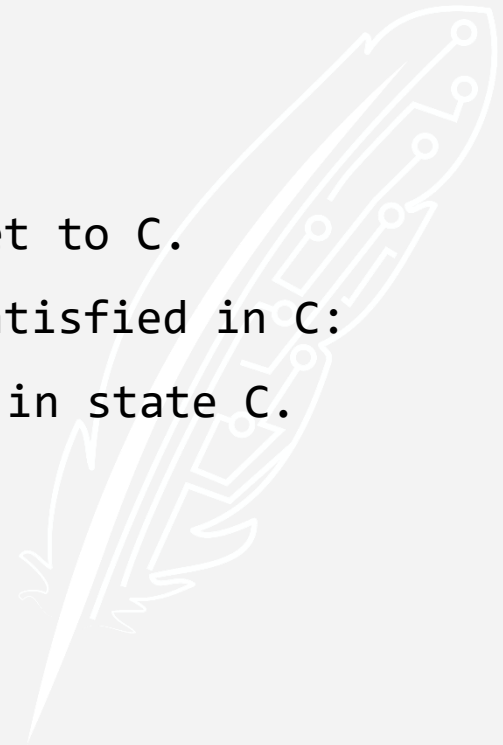
- If C is a goal state, return the plan to get to C .

- For every step S whose preconditions are satisfied in C :

 - Let N be the state after taking step S in state C .

 - Push N onto the queue.

Return failure.



State-Space Search

Priority Queue:



State-Space Search

Start by pushing the initial state onto the priority queue.

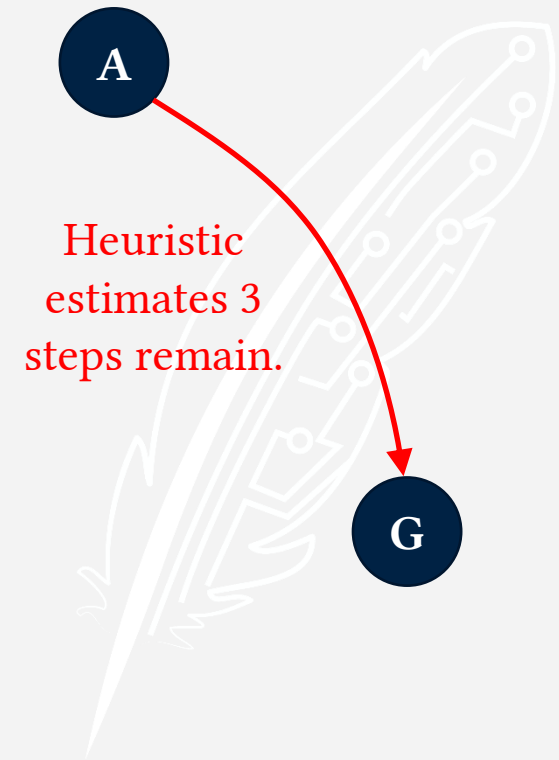
Priority Queue:



State-Space Search

Start by pushing the initial state onto the priority queue.

Priority Queue:

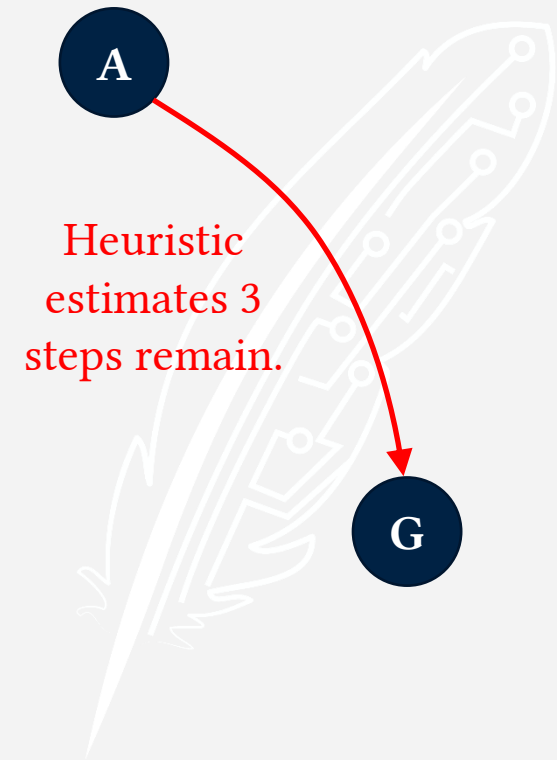


State-Space Search

Start by pushing the initial state onto the priority queue.

Priority Queue:

A: $0 + 3 = 3$



State-Space Search

Pop a state off the queue.

Current State: A

Current Plan: \emptyset

Priority Queue:



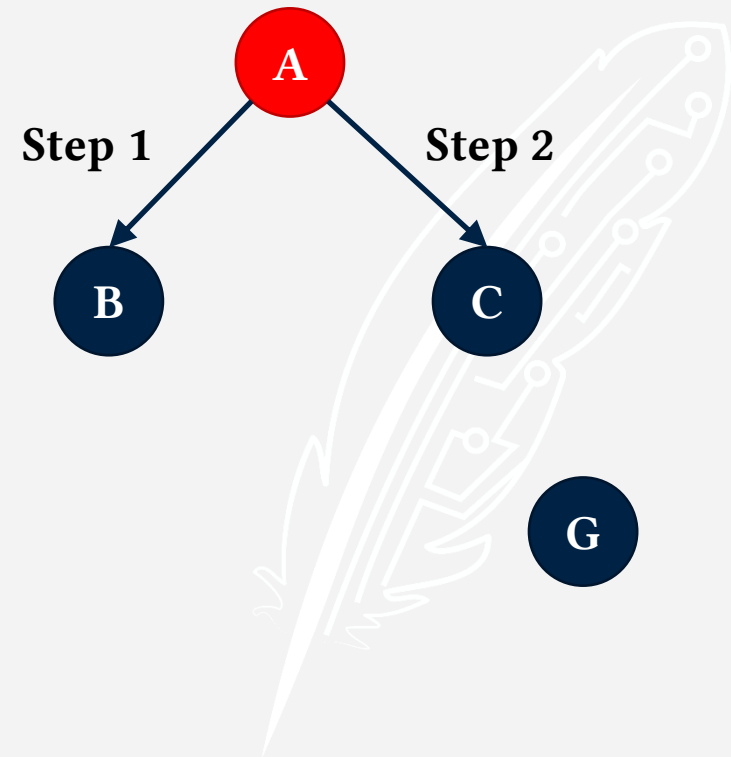
State-Space Search

Expand the current state.

Current State: A

Current Plan: \emptyset

Priority Queue:



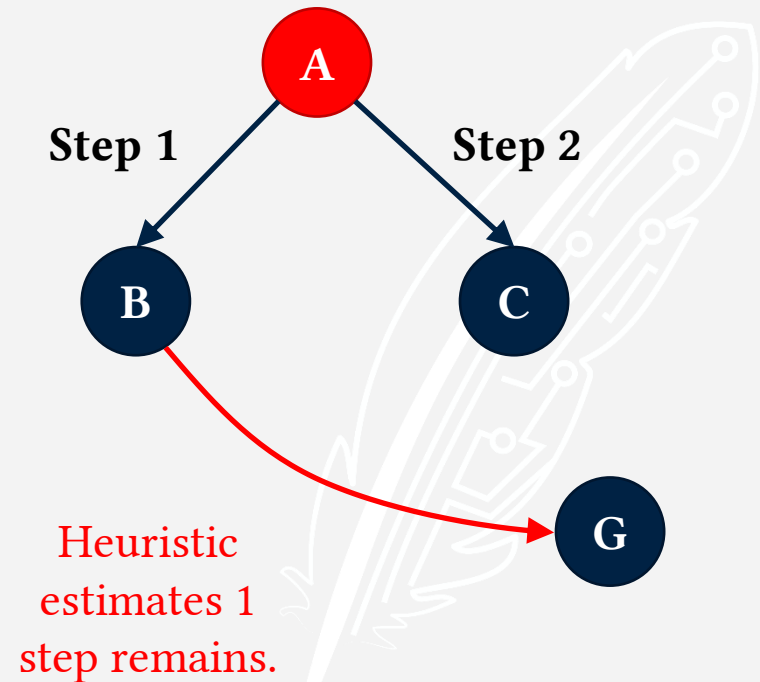
State-Space Search

Put children on the queue.

Current State: A

Current Plan: \emptyset

Priority Queue:



State-Space Search

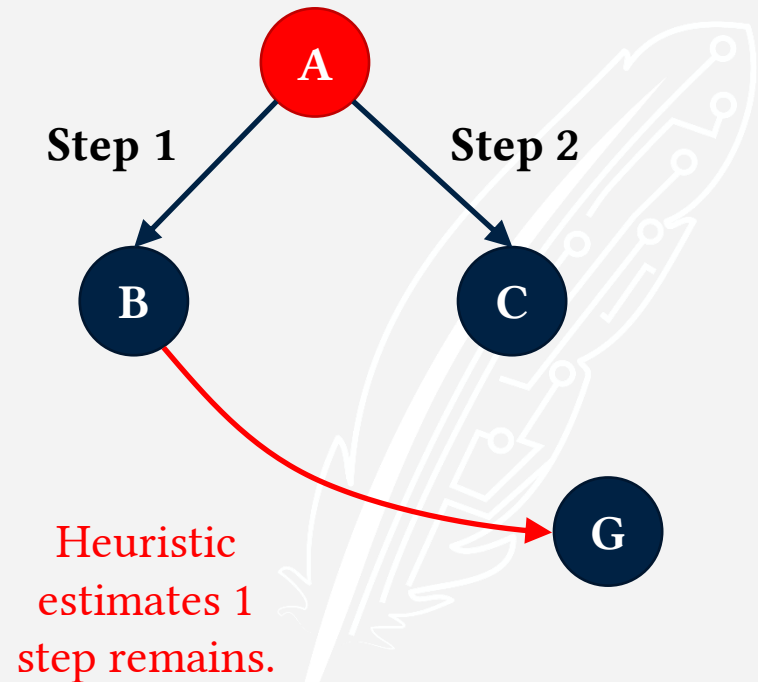
Put children on the queue.

Current State: A

Current Plan: \emptyset

Priority Queue:

B: $1 + 1 = 2$



State-Space Search

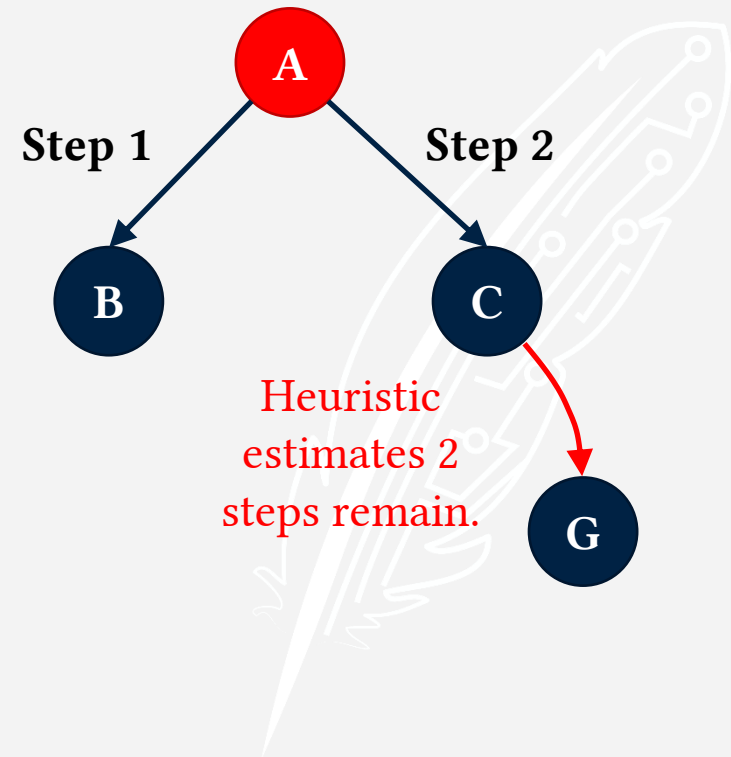
Put children on the queue.

Current State: A

Current Plan: \emptyset

Priority Queue:

B: $1 + 1 = 2$



State-Space Search

Put children on the queue.

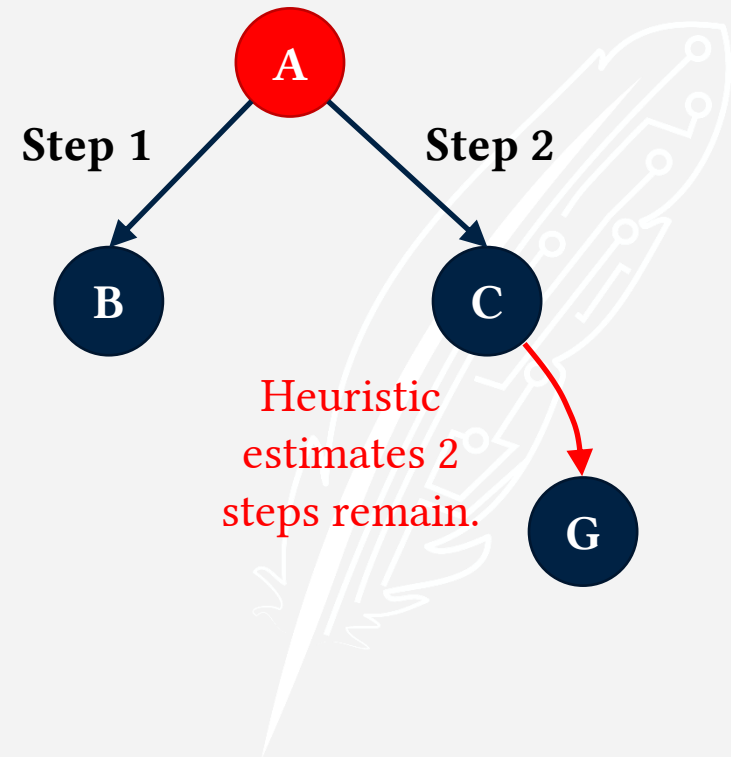
Current State: A

Current Plan: \emptyset

Priority Queue:

B: $1 + 1 = 2$

C: $1 + 2 = 3$



State-Space Search

Put children on the queue.

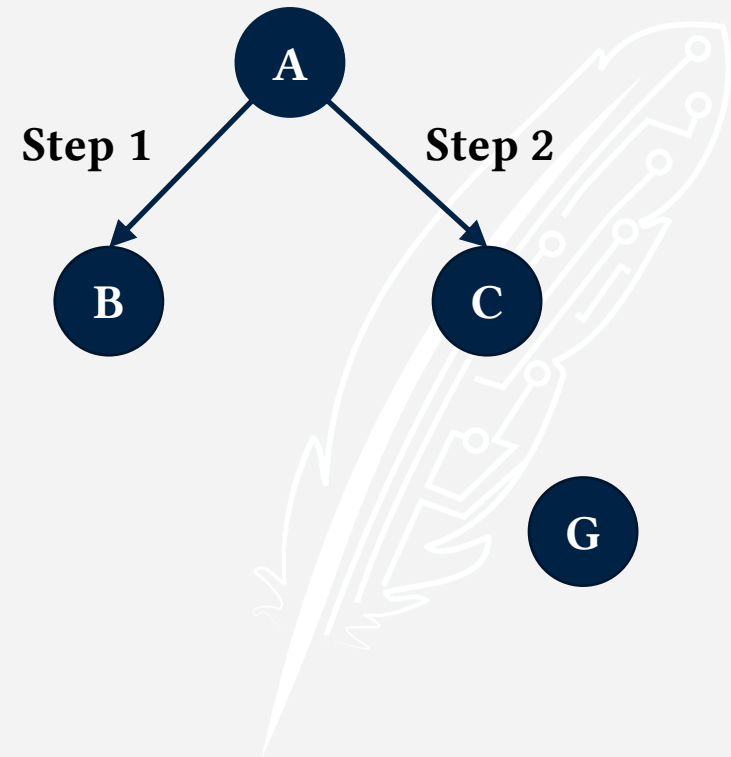
Current State: A

Current Plan: \emptyset

Priority Queue:

B: $1 + 1 = 2$

C: $1 + 2 = 3$



State-Space Search

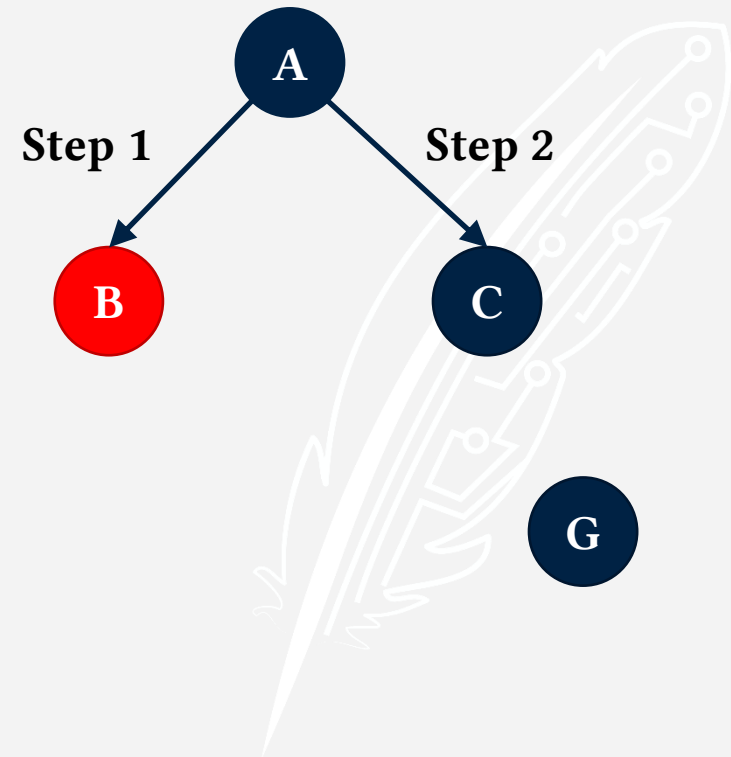
Pop a state off the queue.

Current State: B

Current Plan: 1

Priority Queue:

C: $1 + 2 = 3$



State-Space Search

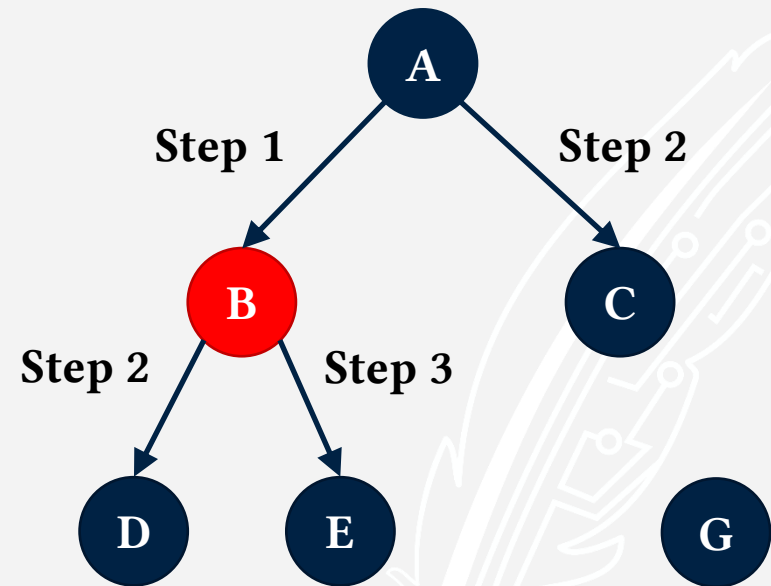
Expand the current state.

Current State: B

Current Plan: 1

Priority Queue:

C: $1 + 2 = 3$



State-Space Search

Put children on the queue.

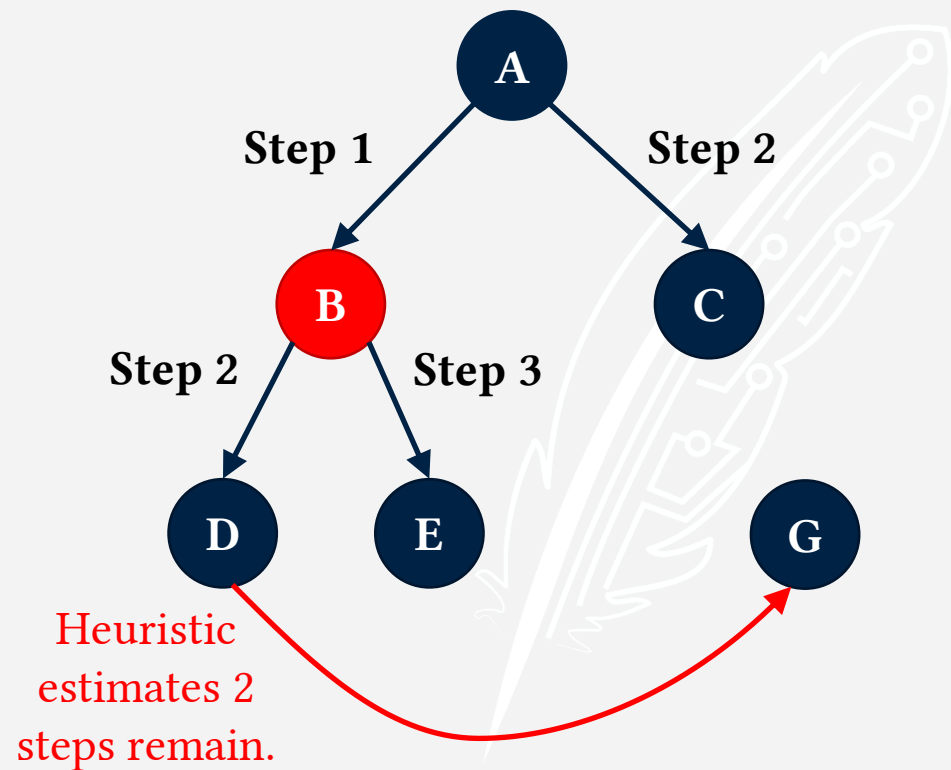
Current State: B

Current Plan: 1

Priority Queue:

C: $1 + 2 = 3$

D: $2 + 2 = 4$



State-Space Search

Put children on the queue.

Current State: B

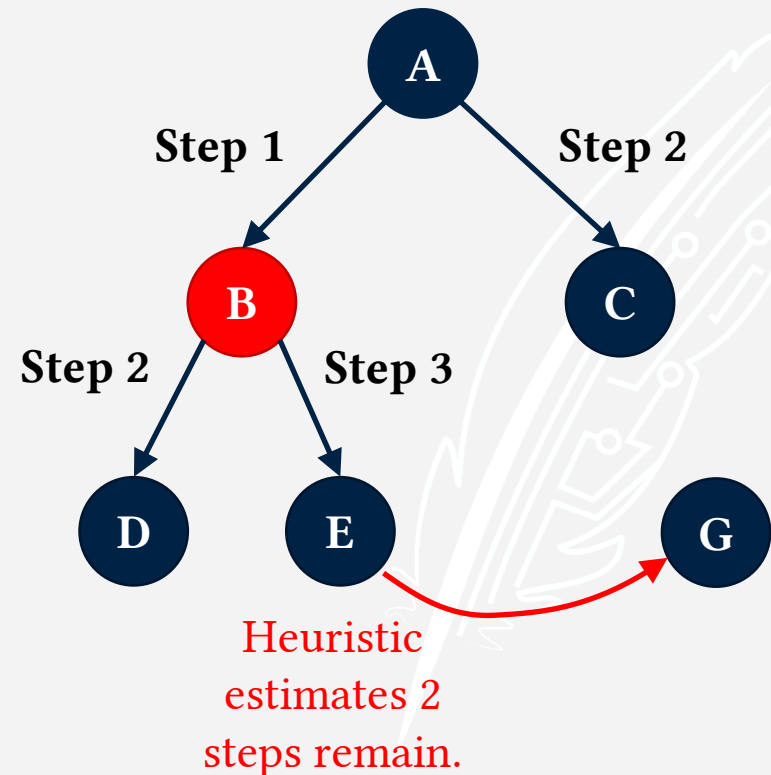
Current Plan: 1

Priority Queue:

C: $1 + 2 = 3$

D: $2 + 2 = 4$

E: $2 + 2 = 4$



State-Space Search

Put children on the queue.

Current State: B

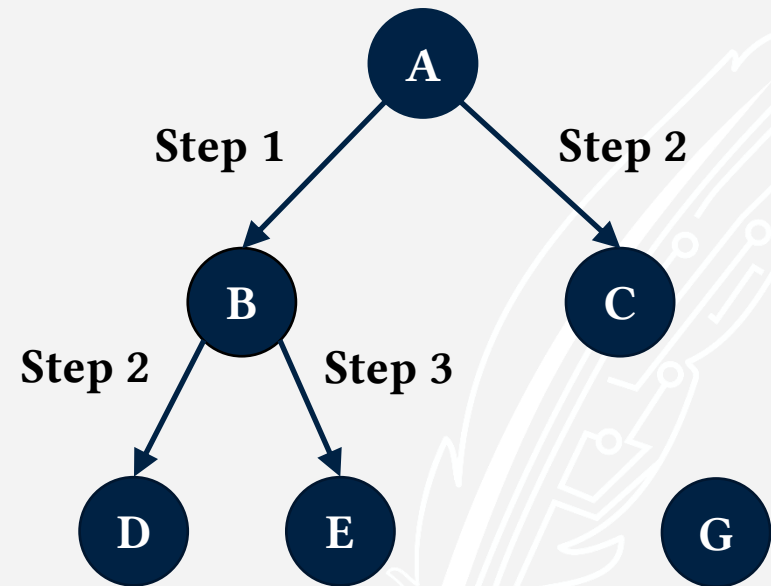
Current Plan: 1

Priority Queue:

C: $1 + 2 = 3$

D: $2 + 2 = 4$

E: $2 + 2 = 4$



State-Space Search

Pop a state off the queue.

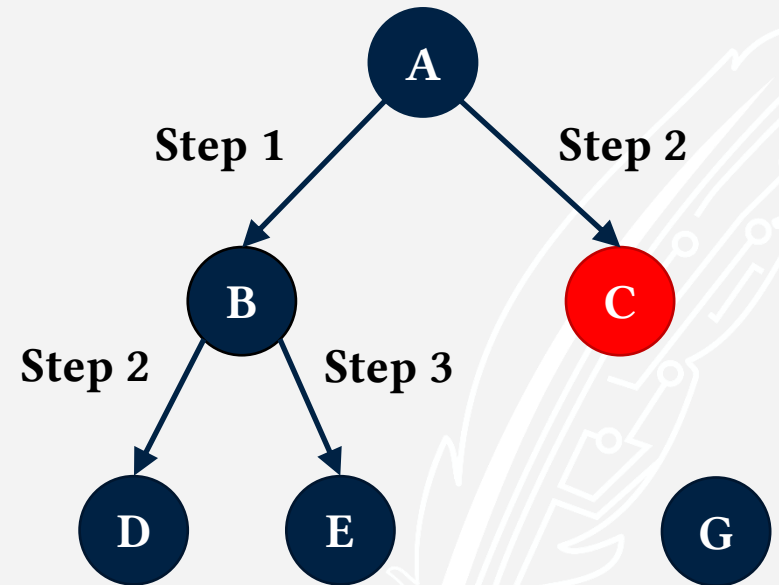
Current State: C

Current Plan: 2

Priority Queue:

D: $2 + 2 = 4$

E: $2 + 2 = 4$



State-Space Search

Expand the current state.

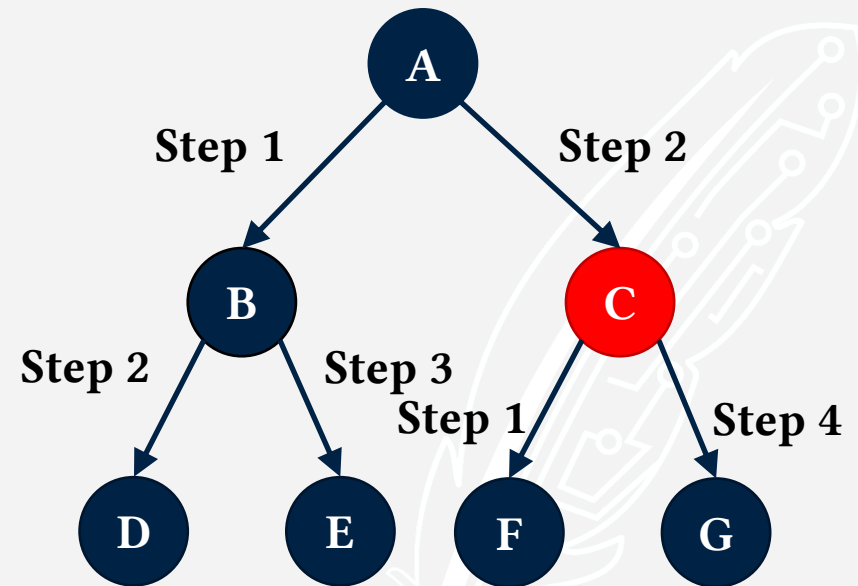
Current State: C

Current Plan: 2

Priority Queue:

D: $2 + 2 = 4$

E: $2 + 2 = 4$



State-Space Search

Put children on the queue.

Current State: C

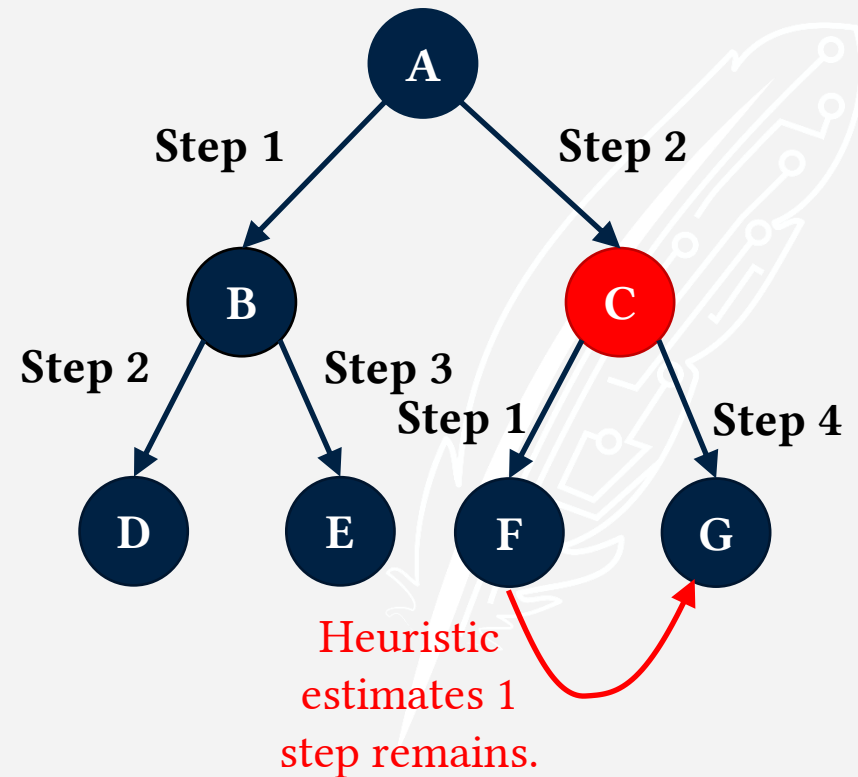
Current Plan: 2

Priority Queue:

F: $2 + 1 = 3$

D: $2 + 2 = 4$

E: $2 + 2 = 4$



State-Space Search

Put children on the queue.

Current State: C

Current Plan: 2

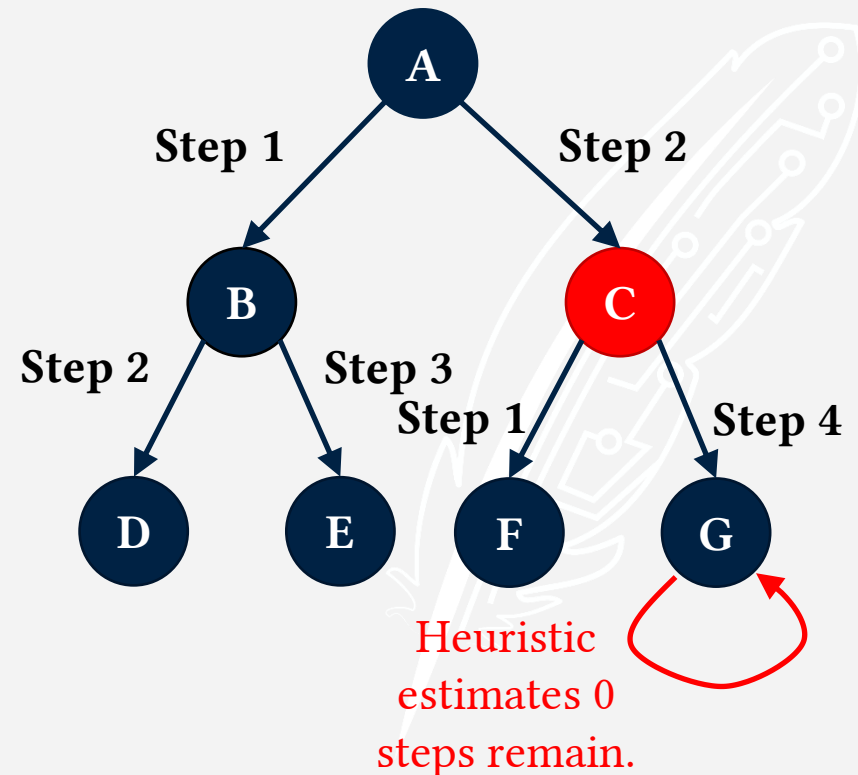
Priority Queue:

G: $2 + 0 = 2$

F: $2 + 1 = 3$

D: $2 + 2 = 4$

E: $2 + 2 = 4$



State-Space Search

Put children on the queue.

Current State: C

Current Plan: 2

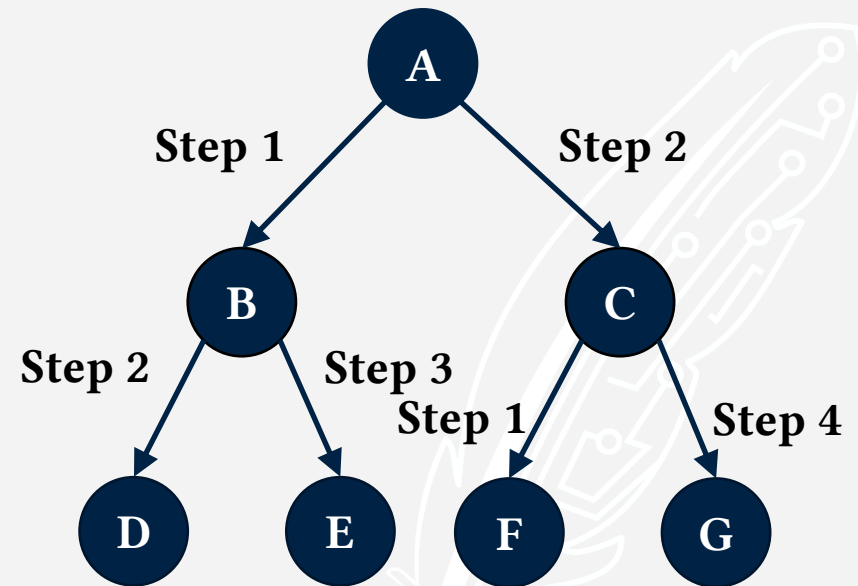
Priority Queue:

G: $2 + 0 = 2$

F: $2 + 1 = 3$

D: $2 + 2 = 4$

E: $2 + 2 = 4$



State-Space Search

Pop a state off the queue.

Current State: G

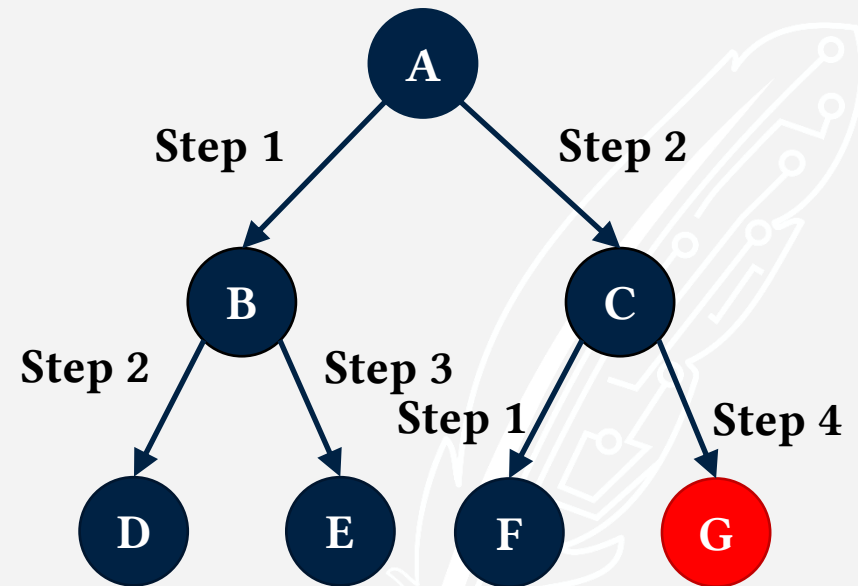
Current Plan: 2, 4

Priority Queue:

F: $2 + 1 = 3$

D: $2 + 2 = 4$

E: $2 + 2 = 4$



State-Space Search

Current state is a goal state!

Current State: G

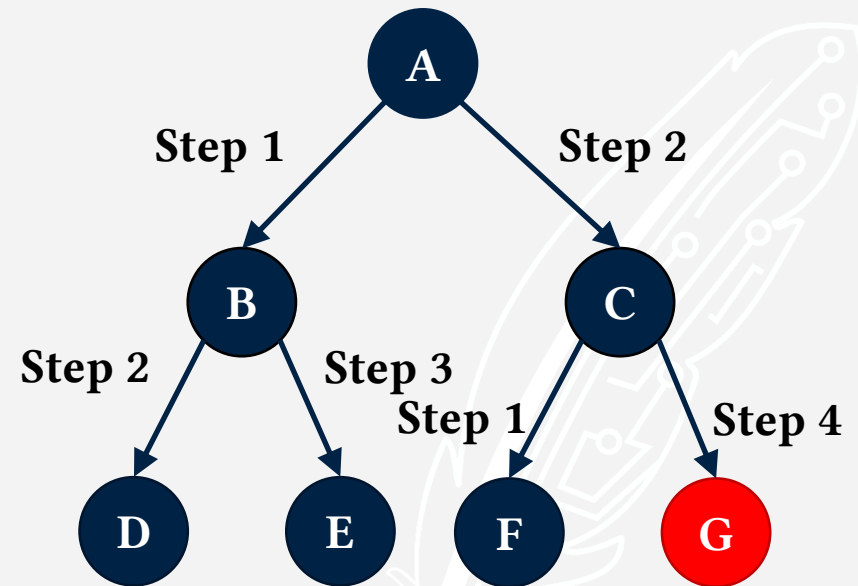
Current Plan: 2, 4

Priority Queue:

F: $2 + 1 = 3$

D: $2 + 2 = 4$

E: $2 + 2 = 4$



State-Space Search

Return plan to reach G.

Current State: G

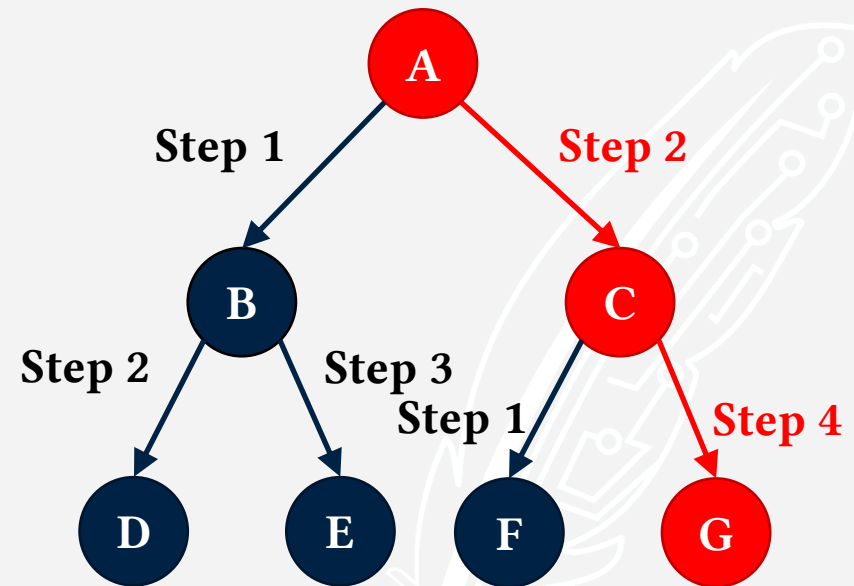
Current Plan: 2, 4

Priority Queue:

F: $2 + 1 = 3$

D: $2 + 2 = 4$

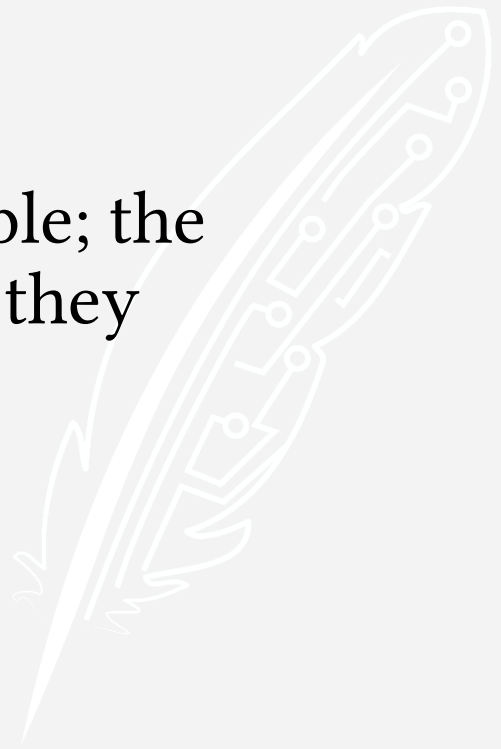
E: $2 + 2 = 4$



Planning Heuristics

The speed of a state-space planner is entirely dependent on its heuristic.

State-space planning algorithms are simple; the complexity and ingenuity comes in how they calculate their heuristics.



Heuristics

A state-space planning heuristic estimates the answer to the following question: “Given some current state, how many more steps need to be taken before a goal state is reached?”

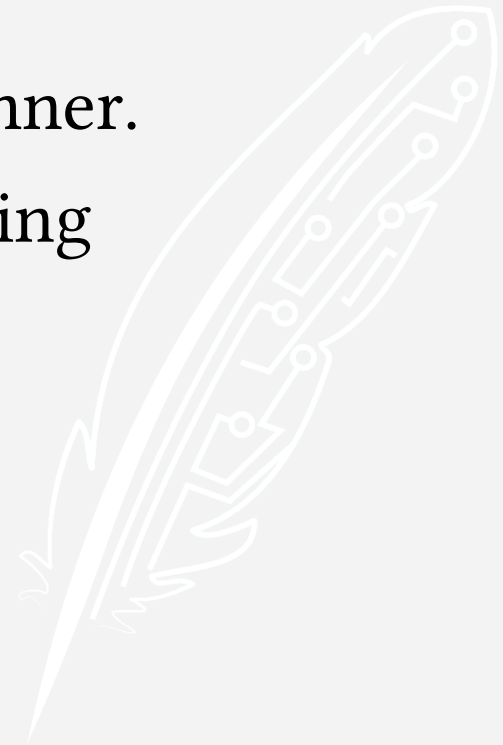
Ideally, a heuristics is:

- Highly accurate
- Admissible
- Fast to calculate



Heuristic Search Planner (HSP)

- Created by Blai Bonet, Gábor Loerincs, and Héctor Geffner
- Perhaps the first viable state-space planner.
- Winner of the first International Planning Competition in 1998



HSP's Heuristic

Input: The current state.

Every literal has a cost, initially ∞ .

Every literal that is true in the current state has a cost of 0.

The cost of a conjunction is the sum of the costs of its conjuncts.

Do this until the costs of the literals stop changing:

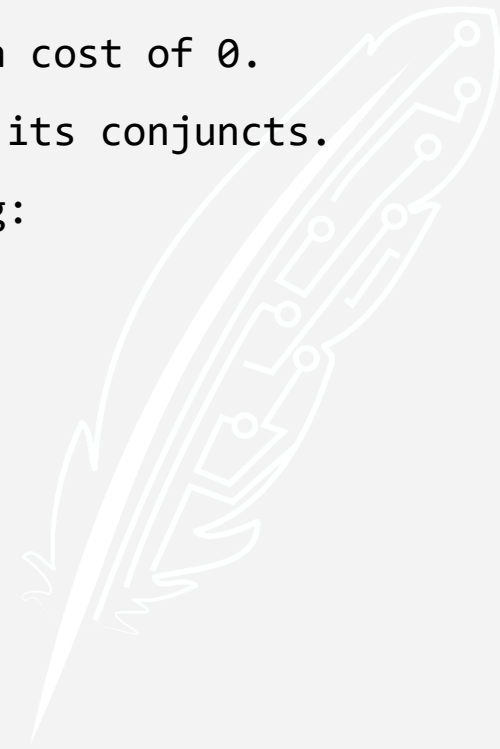
For every step S:

For every literal E in the effect of S:

Let the cost of E be the minimum of:

1. The current cost of E.
2. The cost of S's precondition + 1.

Return the cost of the problem's goal.



HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = \infty$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = \infty$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = \infty$$

$$h(\text{on}(C, A)) = \infty$$

$$h(\neg \text{on}(C, A)) = \infty$$

$$h(\text{clear}(A)) = \infty$$

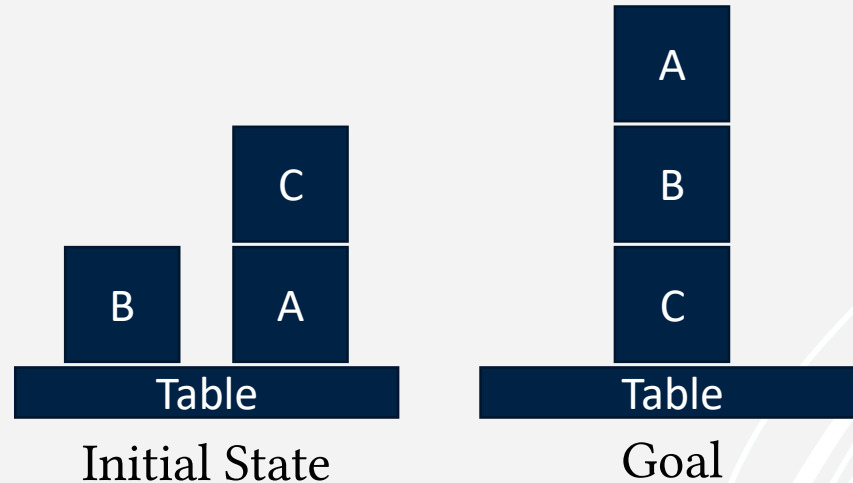
$$h(\text{clear}(B)) = \infty$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = \infty$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$



Note: For the sake of a small example, we will only consider *some* of the literals and actions. When HSP computes its heuristic, it considers *all* literals and *all* actions.

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = \infty$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = \infty$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = \infty$$

$$h(\text{on}(C, A)) = \infty$$

$$h(\neg \text{on}(C, A)) = \infty$$

$$h(\text{clear}(A)) = \infty$$

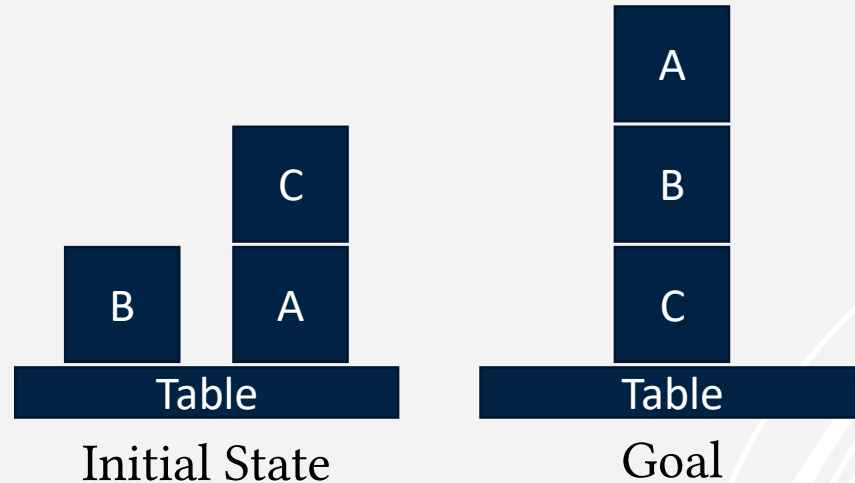
$$h(\text{clear}(B)) = \infty$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = \infty$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$



Start with the cost of every literal set to ∞ .

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = \infty$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = \infty$$

$$h(\text{clear}(A)) = \infty$$

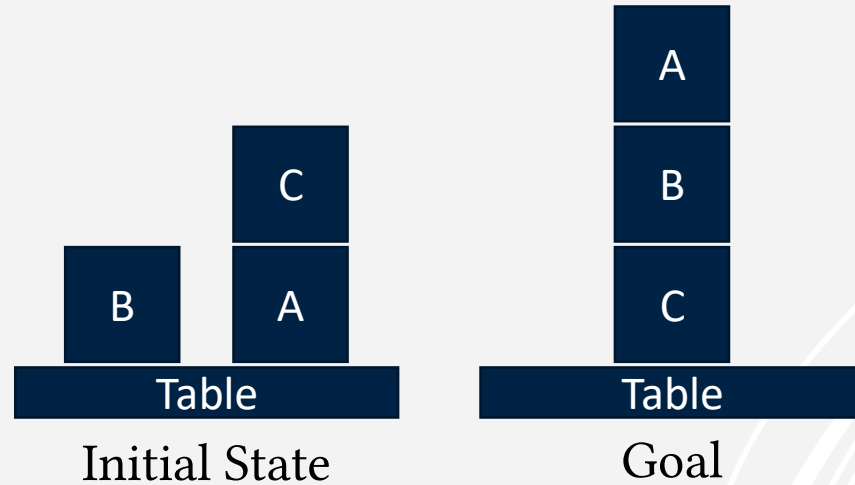
$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$



Set the cost of every literal that is true in the initial state to 0.

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = \infty$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = \infty$$

$$h(\text{clear}(A)) = \infty$$

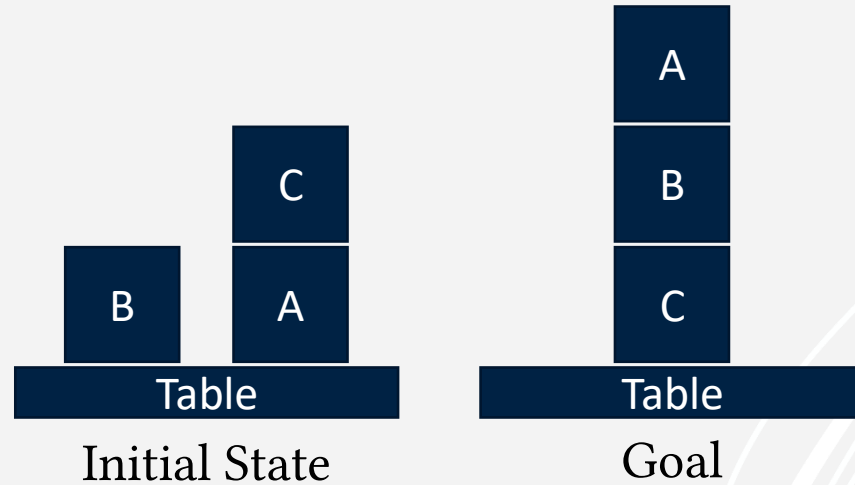
$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$



Update the cost of every effect of every step.

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = \infty$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = \infty$$

$$h(\text{clear}(A)) = \infty$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$

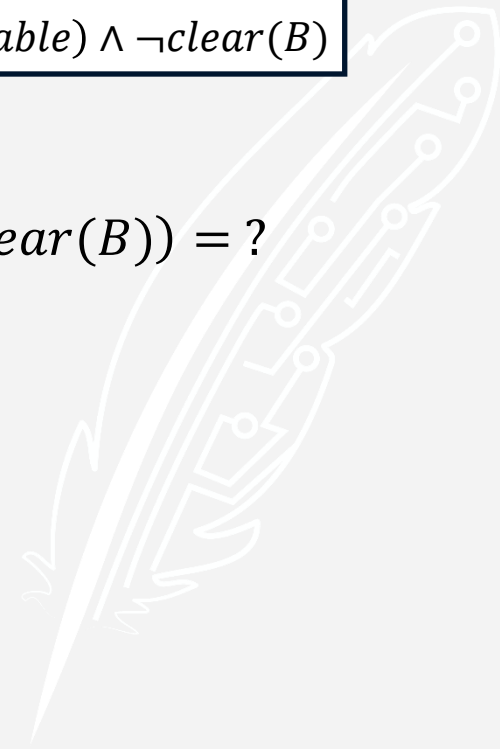
$$\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)$$

$$\text{move}(A, \text{Table}, B)$$

$$\text{on}(A, B) \wedge \neg \text{on}(A, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(B)$$

Cost of precondition:

$$h(\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)) = ?$$



HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = \infty$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = \infty$$

$$h(\text{clear}(A)) = \infty$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$

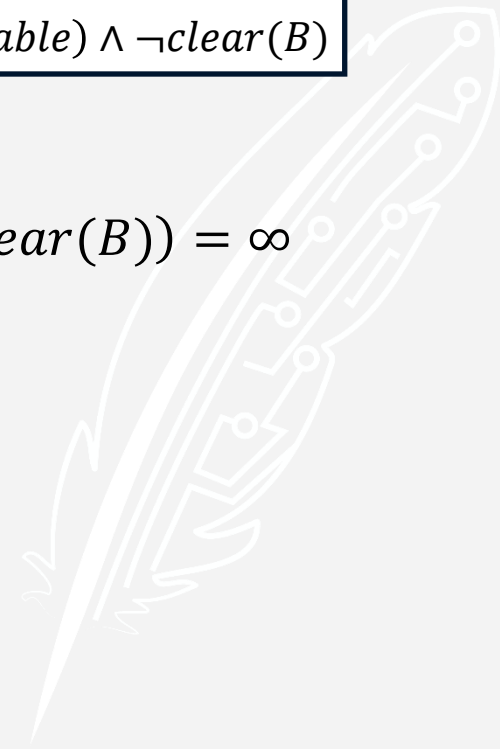
$$\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)$$

$$\text{move}(A, \text{Table}, B)$$

$$\text{on}(A, B) \wedge \neg \text{on}(A, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(B)$$

Cost of precondition:

$$h(\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)) = \infty$$



HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = \infty$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = \infty$$

$$h(\text{clear}(A)) = \infty$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$

$$\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)$$

$$\text{move}(A, \text{Table}, B)$$

$$\text{on}(A, B) \wedge \neg \text{on}(A, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(B)$$

Cost of precondition:

$$h(\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)) = \infty$$

$$\text{Set } h(\text{on}(A, B)) = \min(\infty, \infty + 1)$$

$$\text{Set } h(\neg \text{on}(A, \text{Table})) = \min(\infty, \infty + 1)$$

$$\text{Set } h(\text{clear}(\text{Table})) = \min(\infty, \infty + 1)$$

$$\text{Set } h(\neg \text{clear}(B)) = \min(\infty, \infty + 1)$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = \infty$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = \infty$$

$$h(\text{clear}(A)) = \infty$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$

$$\text{on}(C, A) \wedge \text{clear}(C)$$

$$\text{moveToTable}(C, A)$$

$$\text{on}(C, \text{Table}) \wedge \neg \text{on}(C, A) \wedge \text{clear}(A)$$

Cost of precondition:

$$h(\text{on}(C, A) \wedge \text{clear}(C)) = ?$$



HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = \infty$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = \infty$$

$$h(\text{clear}(A)) = \infty$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$

$\text{on}(C, A) \wedge \text{clear}(C)$

$\text{moveToTable}(C, A)$

$\text{on}(C, \text{Table}) \wedge \neg \text{on}(C, A) \wedge \text{clear}(A)$

Cost of precondition:

$$h(\text{on}(C, A) \wedge \text{clear}(C)) = 0$$

$$\text{Set } h(\text{on}(C, \text{Table})) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\neg \text{on}(C, A)) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\text{clear}(A)) = \min(\infty, 0 + 1)$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$

$$\text{on}(C, A) \wedge \text{clear}(C)$$

$$\text{moveToTable}(C, A)$$

$$\text{on}(C, \text{Table}) \wedge \neg \text{on}(C, A) \wedge \text{clear}(A)$$

Cost of precondition:

$$h(\text{on}(C, A) \wedge \text{clear}(C)) = 0$$

$$\text{Set } h(\text{on}(C, \text{Table})) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\neg \text{on}(C, A)) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\text{clear}(A)) = \min(\infty, 0 + 1)$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$

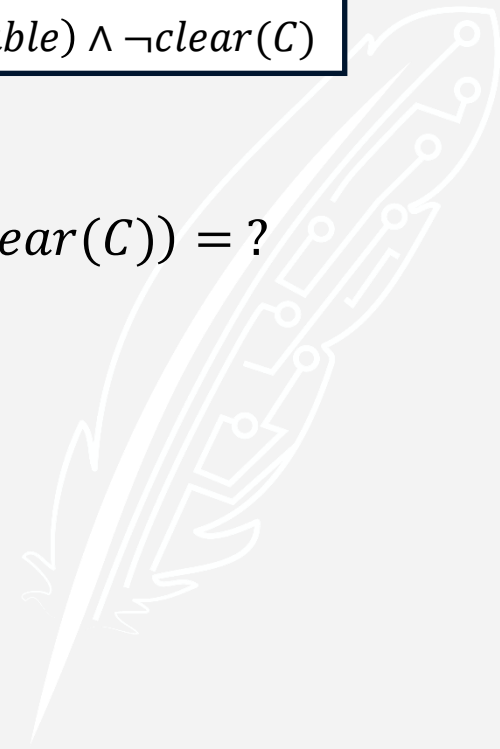
$$\text{on}(B, \text{Table}) \wedge \text{clear}(B) \wedge \text{clear}(C)$$

$$\text{move}(B, \text{Table}, C)$$

$$\text{on}(B, C) \wedge \neg \text{on}(B, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(C)$$

Cost of precondition:

$$h(\text{on}(B, \text{Table}) \wedge \text{clear}(B) \wedge \text{clear}(C)) = ?$$



HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = \infty$$

$$h(\text{on}(B, C)) = \infty$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = \infty$$

$$h(\text{clear}(\text{Table})) = \infty$$

$$\text{on}(B, \text{Table}) \wedge \text{clear}(B) \wedge \text{clear}(C)$$

$$\text{move}(B, \text{Table}, C)$$

$$\text{on}(B, C) \wedge \neg \text{on}(B, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(C)$$

Cost of precondition:

$$h(\text{on}(B, \text{Table}) \wedge \text{clear}(B) \wedge \text{clear}(C)) = 0$$

$$\text{Set } h(\text{on}(B, C)) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\neg \text{on}(B, \text{Table})) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\text{clear}(\text{Table})) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\neg \text{clear}(C)) = \min(\infty, 0 + 1)$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$

$$\text{on}(B, \text{Table}) \wedge \text{clear}(B) \wedge \text{clear}(C)$$

$$\text{move}(B, \text{Table}, C)$$

$$\text{on}(B, C) \wedge \neg \text{on}(B, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(C)$$

Cost of precondition:

$$h(\text{on}(B, \text{Table}) \wedge \text{clear}(B) \wedge \text{clear}(C)) = 0$$

$$\text{Set } h(\text{on}(B, C)) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\neg \text{on}(B, \text{Table})) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\text{clear}(\text{Table})) = \min(\infty, 0 + 1)$$

$$\text{Set } h(\neg \text{clear}(C)) = \min(\infty, 0 + 1)$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

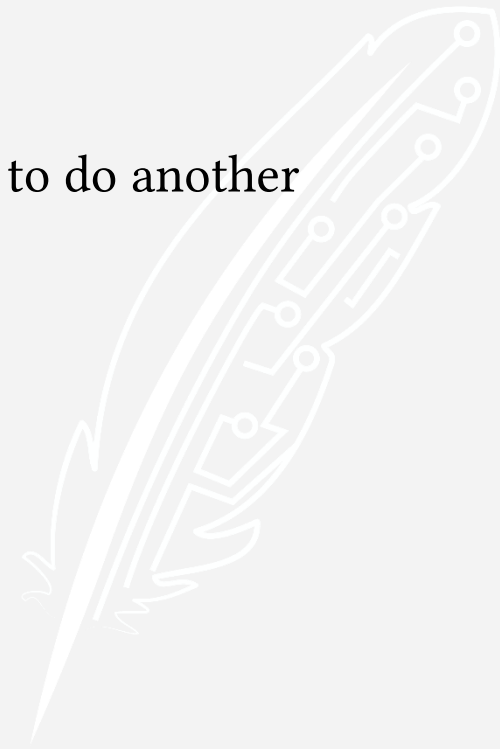
$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$

Some costs changed, so we need to do another round of updates.



HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$

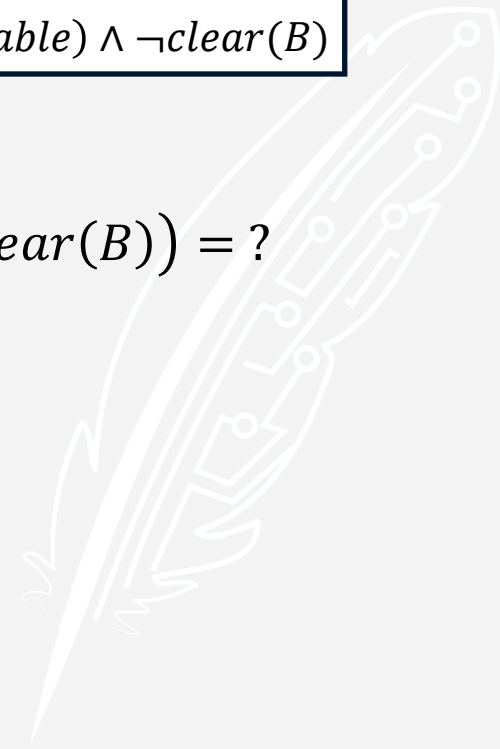
$$\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)$$

$$\text{move}(A, \text{Table}, B)$$

$$\text{on}(A, B) \wedge \neg \text{on}(A, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(B)$$

Cost of precondition:

$$h(\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)) = ?$$



HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = \infty$$

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

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = \infty$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$

$$\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)$$

$$\text{move}(A, \text{Table}, B)$$

$$\text{on}(A, B) \wedge \neg \text{on}(A, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(B)$$

Cost of precondition:

$$h(\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)) = 1$$

$$\text{Set } h(\text{on}(A, B)) = \min(\infty, 1 + 1)$$

$$\text{Set } h(\neg \text{on}(A, \text{Table})) = \min(\infty, 1 + 1)$$

$$\text{Set } h(\text{clear}(\text{Table})) = \min(1, 1 + 1)$$

$$\text{Set } h(\neg \text{clear}(B)) = \min(\infty, 1 + 1)$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = 2$$

$$h(\text{on}(A, B)) = 2$$

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = 2$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$

$$\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)$$

$$\text{move}(A, \text{Table}, B)$$

$$\text{on}(A, B) \wedge \neg \text{on}(A, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(B)$$

Cost of precondition:

$$h(\text{on}(A, \text{Table}) \wedge \text{clear}(A) \wedge \text{clear}(B)) = 1$$

$$\text{Set } h(\text{on}(A, B)) = \min(\infty, 1 + 1)$$

$$\text{Set } h(\neg \text{on}(A, \text{Table})) = \min(\infty, 1 + 1)$$

$$\text{Set } h(\text{clear}(\text{Table})) = \min(1, 1 + 1)$$

$$\text{Set } h(\neg \text{clear}(B)) = \min(\infty, 1 + 1)$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = 2$$

$$h(\text{on}(A, B)) = 2$$

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = 2$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$

$\text{on}(C, A) \wedge \text{clear}(C)$

$\text{moveToTable}(C, A)$

$\text{on}(C, \text{Table}) \wedge \neg \text{on}(C, A) \wedge \text{clear}(A)$

Cost of precondition:

$$h(\text{on}(C, A) \wedge \text{clear}(C)) = 0$$

$$\text{Set } h(\text{on}(C, \text{Table})) = \min(1, 0 + 1)$$

$$\text{Set } h(\neg \text{on}(C, A)) = \min(1, 0 + 1)$$

$$\text{Set } h(\text{clear}(A)) = \min(1, 0 + 1)$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = 2$$

$$h(\text{on}(A, B)) = 2$$

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = 2$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$

$$\text{on}(B, \text{Table}) \wedge \text{clear}(B) \wedge \text{clear}(C)$$

$$\text{move}(B, \text{Table}, C)$$

$$\text{on}(B, C) \wedge \neg \text{on}(B, \text{Table}) \wedge \text{clear}(\text{Table}) \wedge \neg \text{clear}(C)$$

Cost of precondition:

$$h(\text{on}(B, \text{Table}) \wedge \text{clear}(B) \wedge \text{clear}(C)) = 0$$

$$\text{Set } h(\text{on}(B, C)) = \min(1, 0 + 1)$$

$$\text{Set } h(\neg \text{on}(B, \text{Table})) = \min(1, 0 + 1)$$

$$\text{Set } h(\text{clear}(\text{Table})) = \min(1, 0 + 1)$$

$$\text{Set } h(\neg \text{clear}(C)) = \min(1, 0 + 1)$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = 2$$

$$h(\text{on}(A, B)) = 2$$

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = 2$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$

Some costs changed, so we need to do another round of updates.

Spoiler: Nothing will change this round.

After a round where no weights change, we are done.

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = 2$$

$$h(\text{on}(A, B)) = 2$$

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

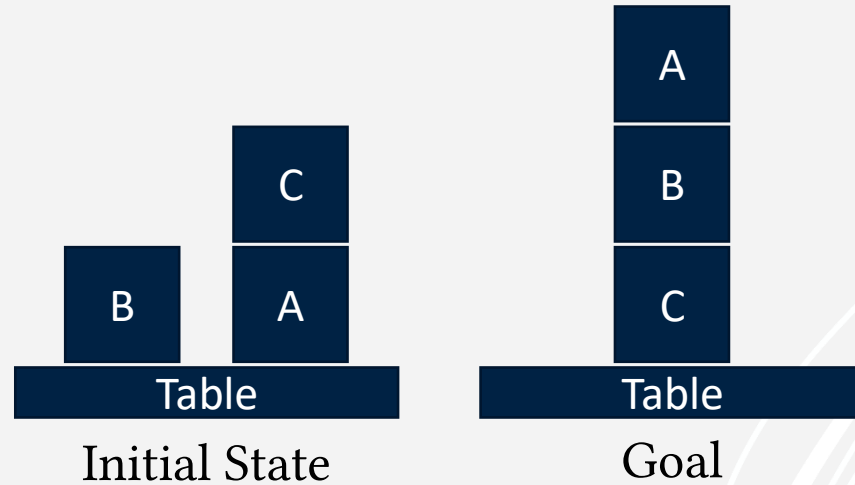
$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = 2$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$



What is the estimated cost of the goal?

$$h(\text{on}(A, B) \wedge \text{on}(B, C)) = ?$$

HSP in Blocks World

Weights:

$$h(\text{on}(A, \text{Table})) = 0$$

$$h(\neg \text{on}(A, \text{Table})) = 2$$

$$h(\text{on}(A, B)) = 2$$

$$h(\text{on}(B, \text{Table})) = 0$$

$$h(\neg \text{on}(B, \text{Table})) = 1$$

$$h(\text{on}(B, C)) = 1$$

$$h(\text{on}(C, \text{Table})) = 1$$

$$h(\text{on}(C, A)) = 0$$

$$h(\neg \text{on}(C, A)) = 1$$

$$h(\text{clear}(A)) = 1$$

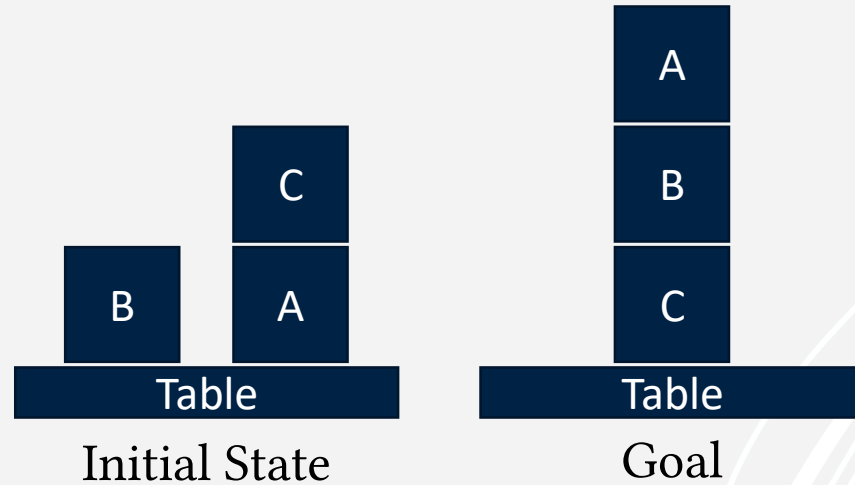
$$h(\text{clear}(B)) = 0$$

$$h(\neg \text{clear}(B)) = 2$$

$$h(\text{clear}(C)) = 0$$

$$h(\neg \text{clear}(C)) = 1$$

$$h(\text{clear}(\text{Table})) = 1$$



What is the estimated cost of the goal?

$$h(\text{on}(A, B) \wedge \text{on}(B, C)) = 2 + 1 = 3$$

HSP's Heuristic

Every literal has a cost, initially ∞ .

Every literal that is true in the initial state has a cost of 0.

The cost of a conjunction is the sum of the costs of its conjuncts.

Do this until the costs of the literals stop changing:

For every step S :

For every literal E in the effect of S :

Let the cost of E be the minimum of:

1. The current cost of E .
2. The cost of S 's precondition + 1.



Planning Problems

During planning, goals can:

- **Interfere:** Progress toward one goal undoes progress toward another goal.
- **Synergize:** Progress toward one goal also makes progress toward another goal.



HSP's Heuristic

- HSP does not account for interference.
(Sometimes called the “ignore delete list” assumption)
- HSP does not account for synergy.
(Because the cost of a goal is the sum of its parts)
- HSP may overestimate, and thus is not admissible.
- In practice, HSP was the first heuristic that was accurate enough to allow for state-space planning.
- Heuristic is efficient to compute.

Analyzing HSP's Heuristic

- The heuristic is based on a relaxed version of the planning problem that is much easier to solve but still provides a good approximation of the original.
- Relaxed problem: We can never get farther from a goal, only closer (i.e. ignore interference).
- The cost of achieving some literal is $1 +$ the cost of achieving the precondition of any step which has that literal as an effect.

Fast-Forward (FF)

- Created by Bernhard Nebel and Jörg Hoffmann
- Top performer in the second (2000) and third (2002) International Planning Competitions
- Observation: HSP's estimates are very similar to those obtained from a plan graph
- Idea: Use a plan graph to estimate the difficulty of a goal and to find a solution to the relaxed problem
- Benefit: Plan graphs account for synergy in goals

FF Heuristic

Given a current state S and a goal G :

Construct a plan graph such that layer 0 is S .

Extend the plan graph until all goals in G appear.

(Do not use persistence steps or mutexes.)

Extract a solution as graphplan does.

Return the size of the resulting plan.



Cargo Problem

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

Goal: $at(C1, MSY) \wedge at(C2, MSY)$

Plan:

1. $load(C1, P1, ATL)$
2. $load(C2, P1, ATL)$
3. $fly(P1, ATL, MSY)$
4. $unload(C1, P1, MSY)$
5. $unload(C2, P1, MSY)$

HSP Estimate: ? steps

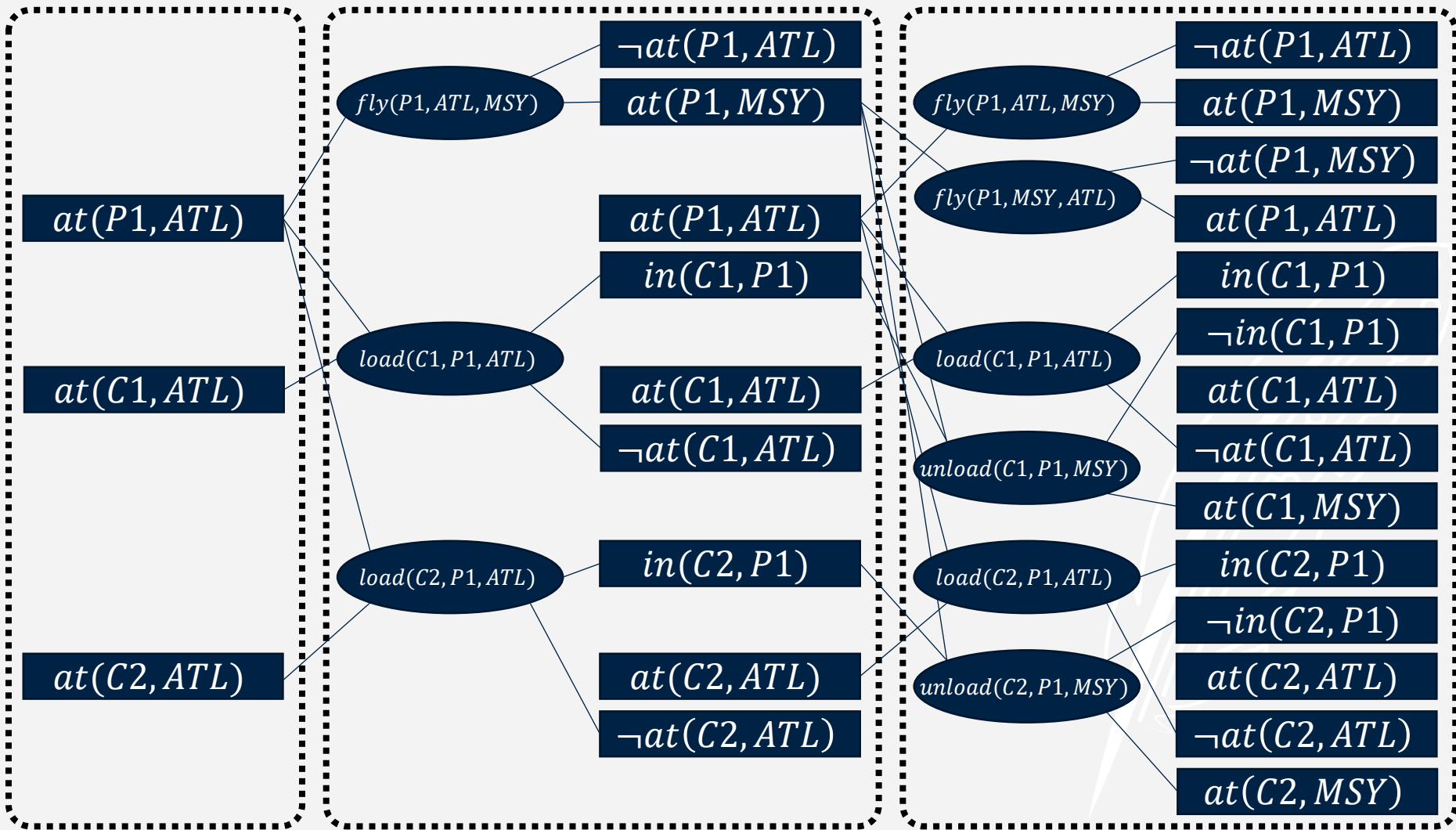
Plan Graph Estimate: ? steps

FF Estimate: ? steps

Level 0

Level 1

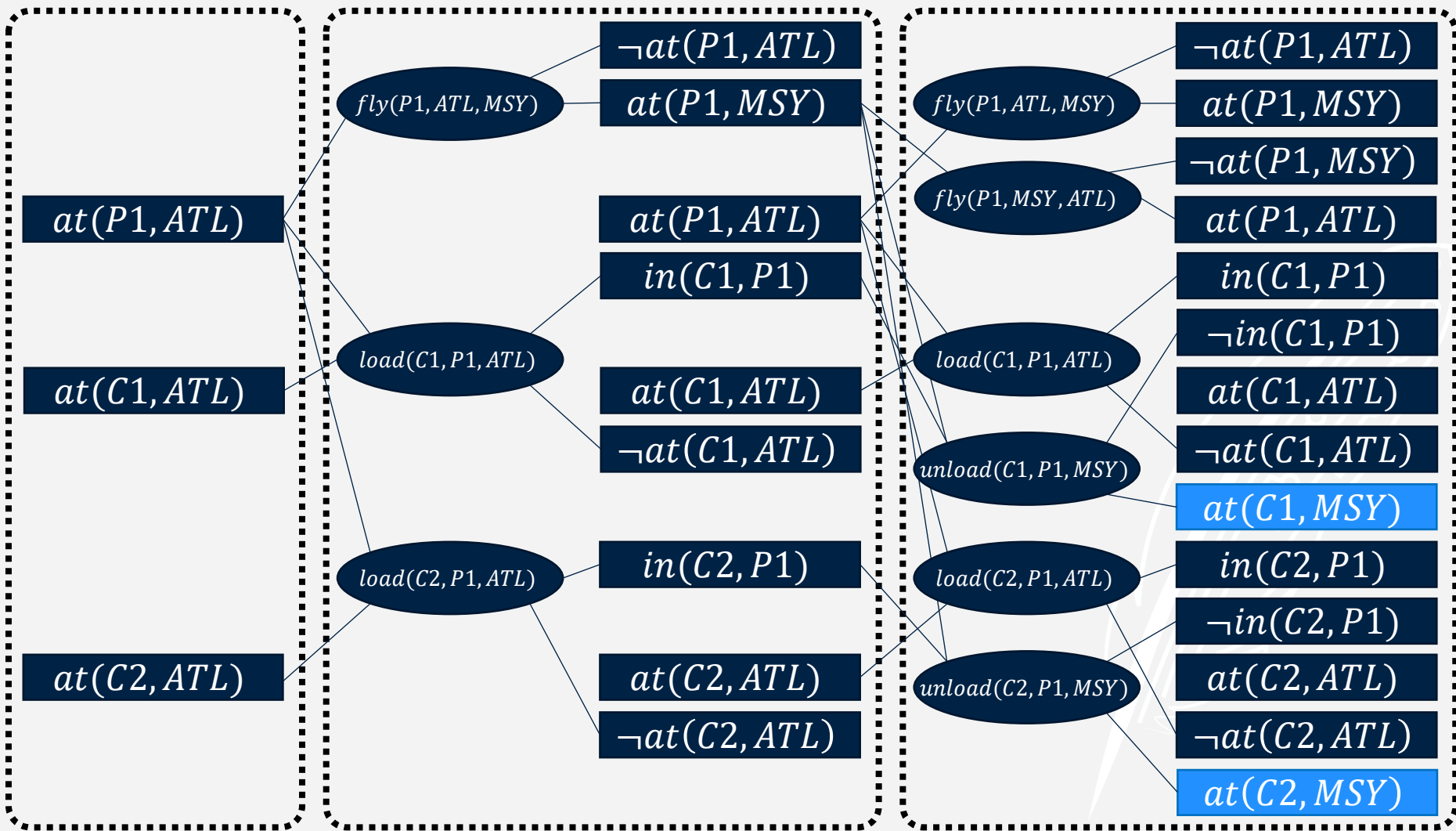
Level 2



Level 0

Level 1

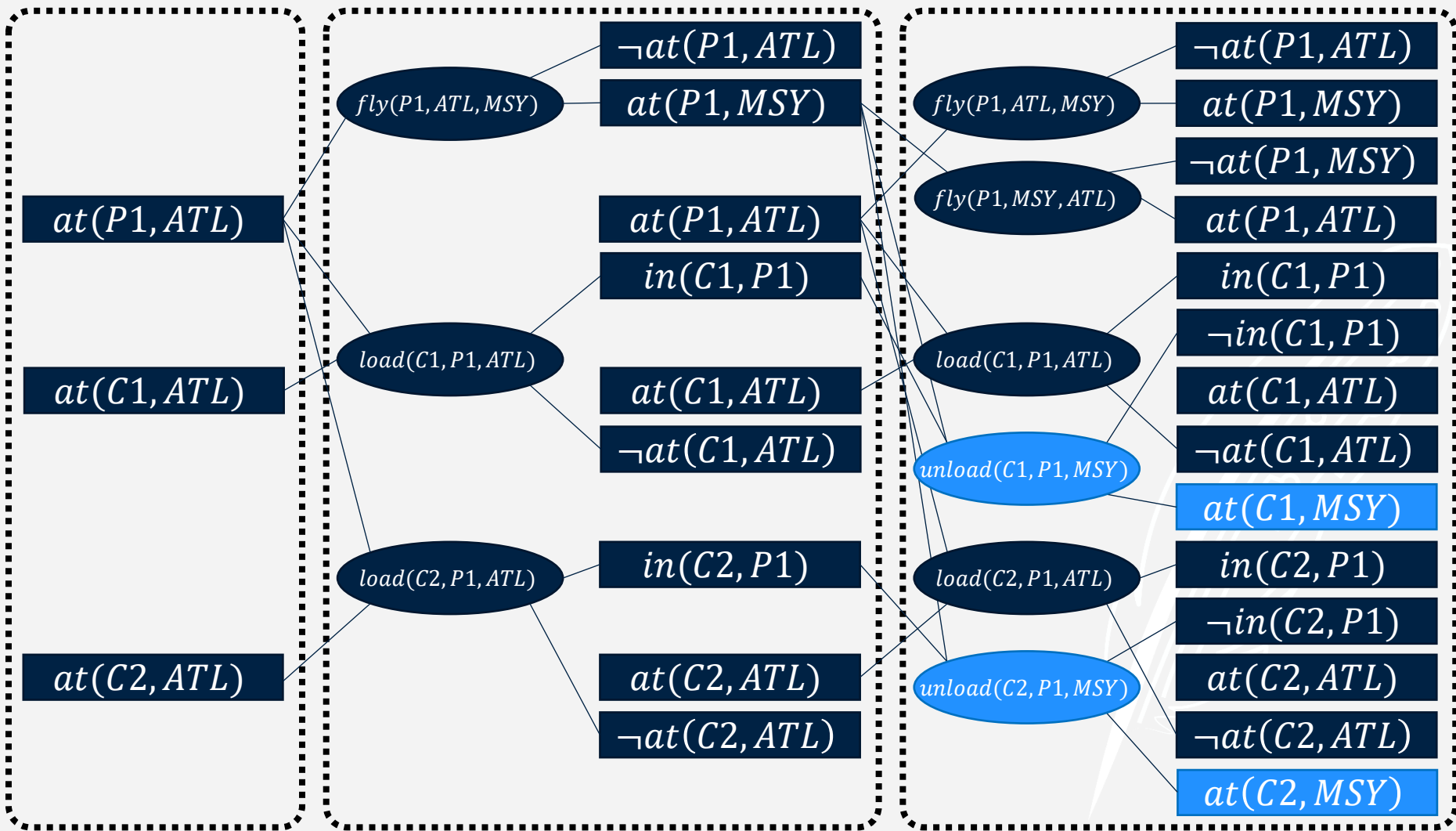
Level 2



Level 0

Level 1

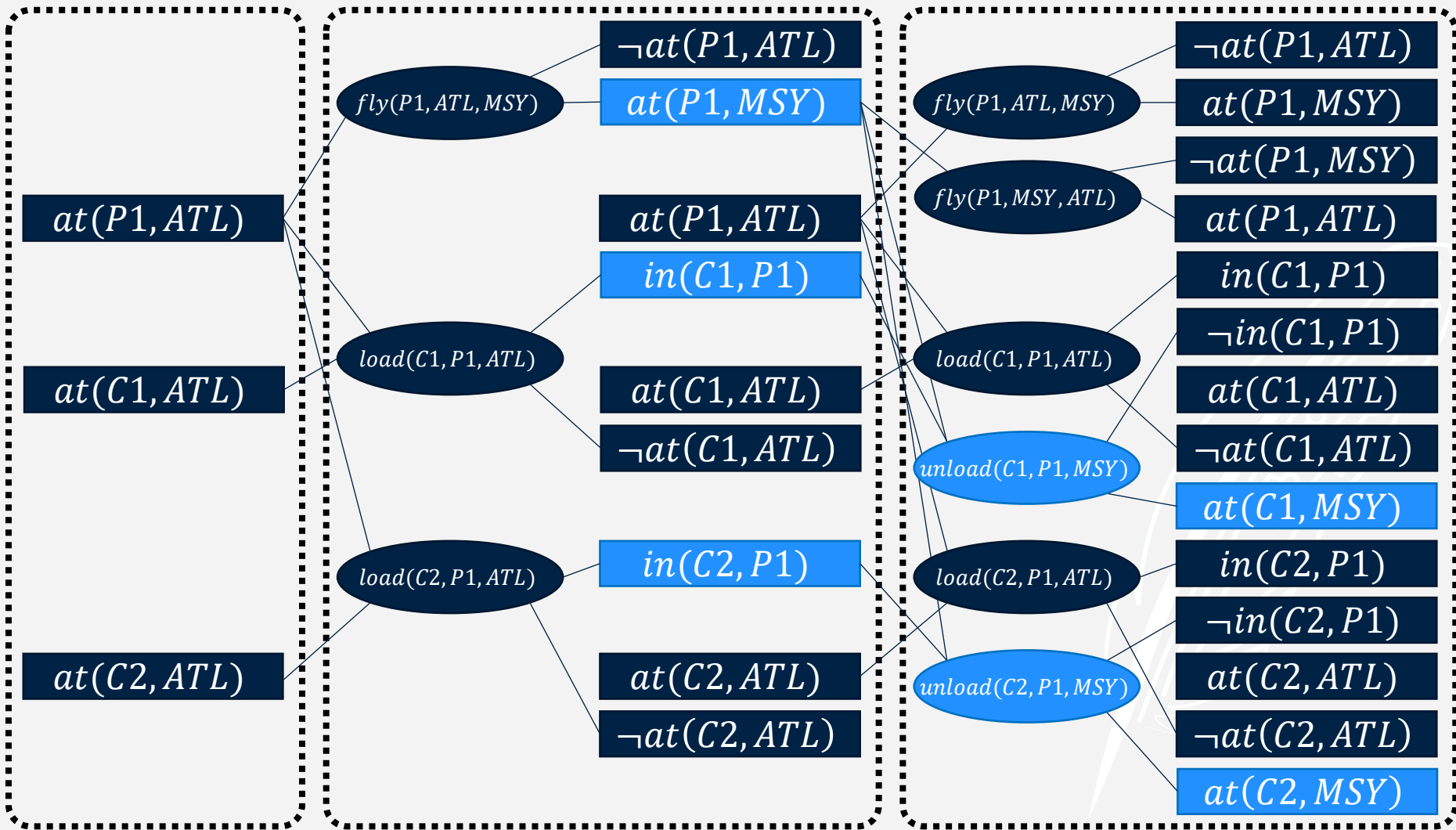
Level 2



Level 0

Level 1

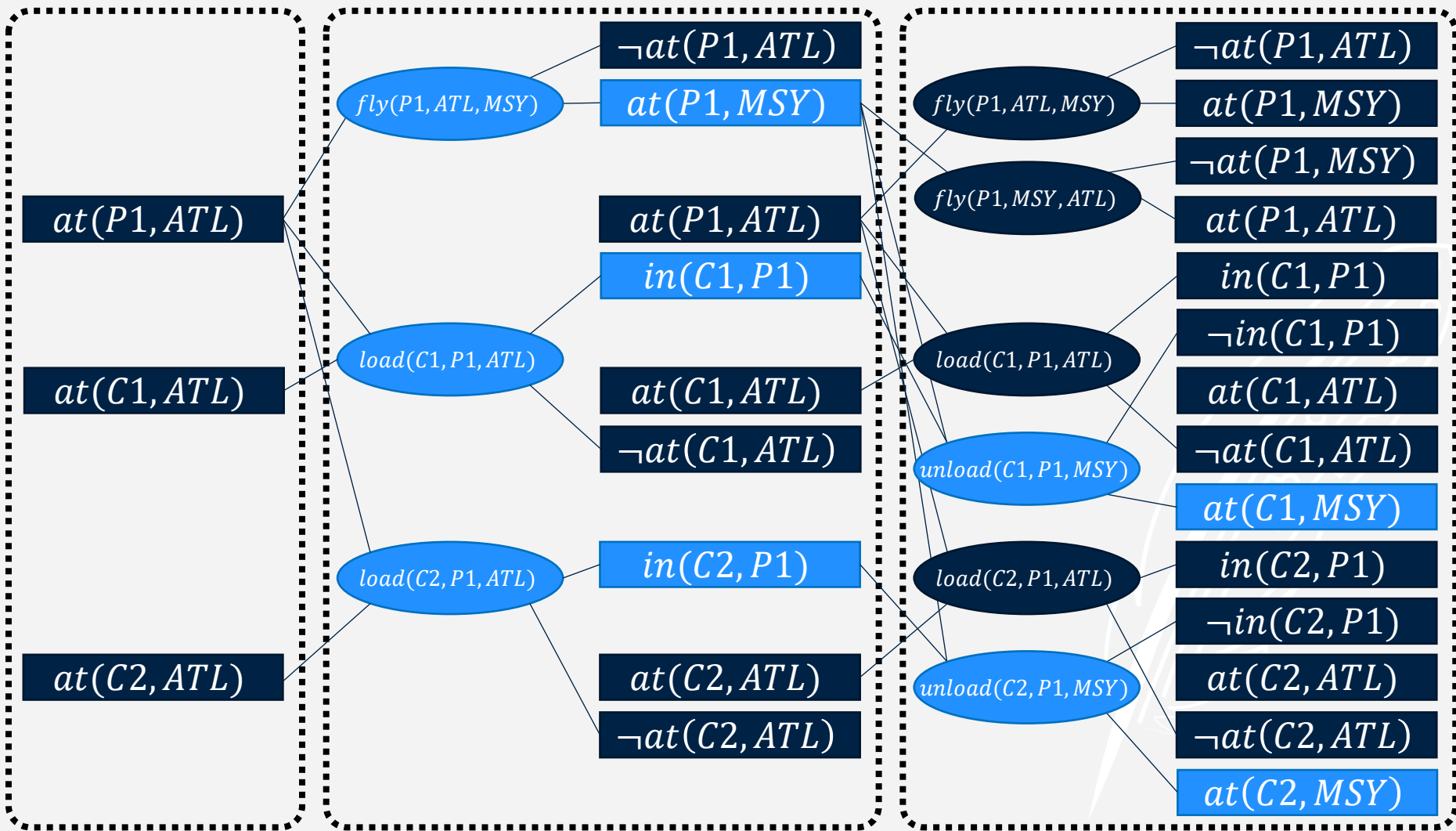
Level 2



Level 0

Level 1

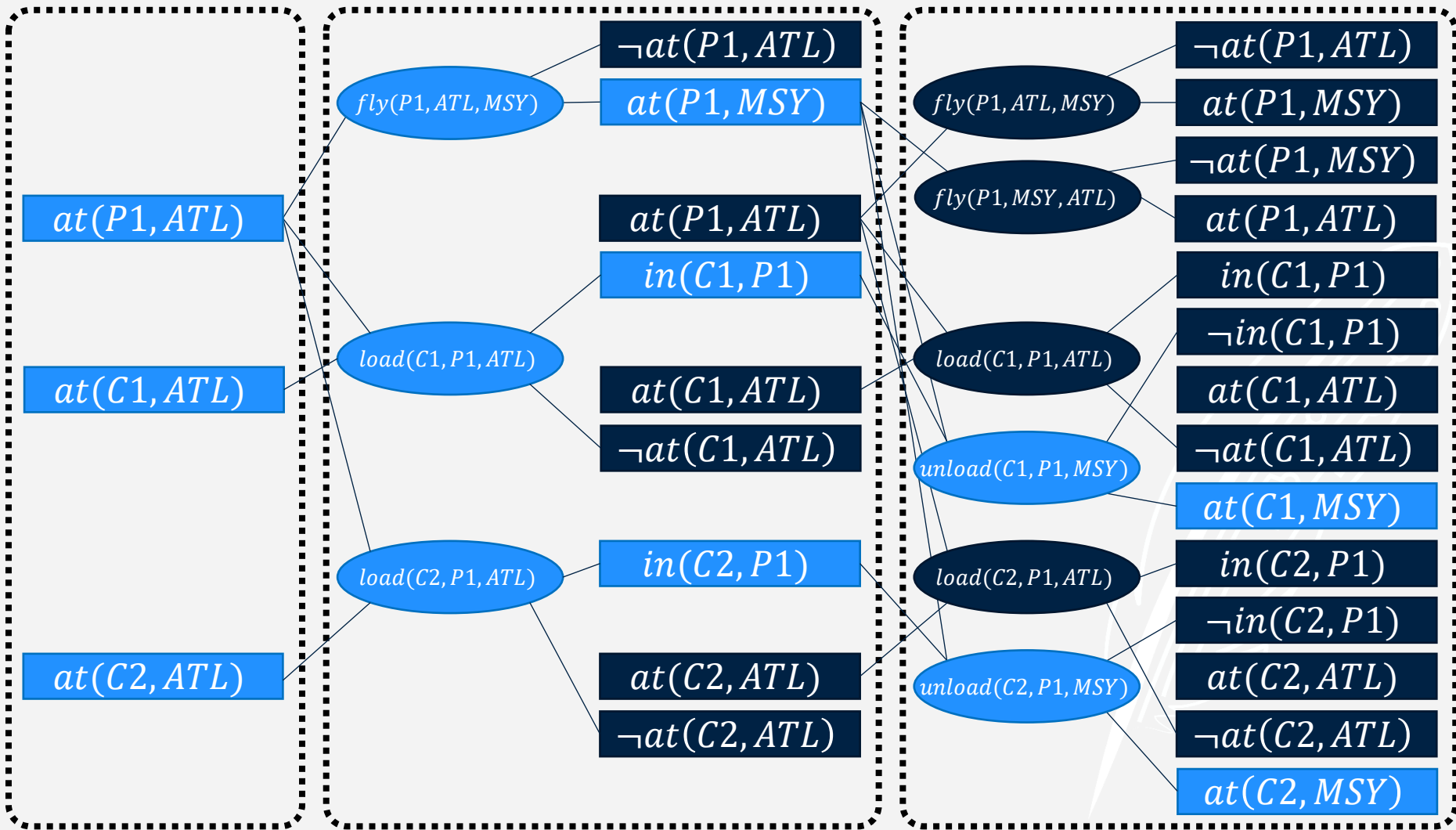
Level 2



Level 0

Level 1

Level 2



Cargo Problem

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

Goal: $at(C1, MSY) \wedge at(C2, MSY)$

Plan:

1. $load(C1, P1, ATL)$
2. $load(C2, P1, ATL)$
3. $fly(P1, ATL, MSY)$
4. $unload(C1, P1, MSY)$
5. $unload(C2, P1, MSY)$

HSP Estimate: ? steps

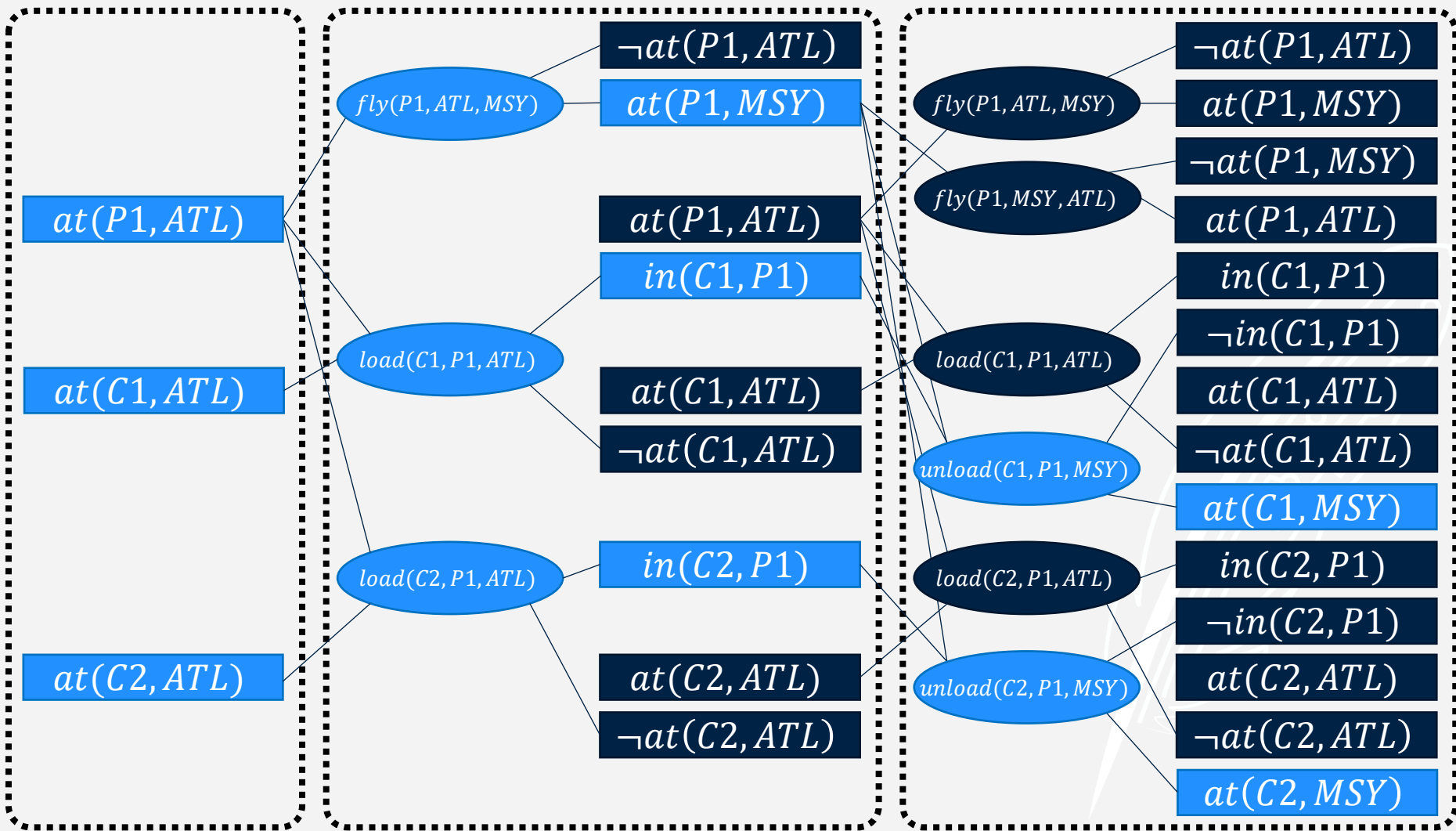
Plan Graph Estimate: ? steps

FF Estimate: ? steps

Level 0

Level 1

Level 2



Cargo Problem

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

Goal: $at(C1, MSY) \wedge at(C2, MSY)$

Plan:

1. $load(C1, P1, ATL)$
2. $load(C2, P1, ATL)$
3. $fly(P1, ATL, MSY)$
4. $unload(C1, P1, MSY)$
5. $unload(C2, P1, MSY)$

HSP Estimate: 6 steps

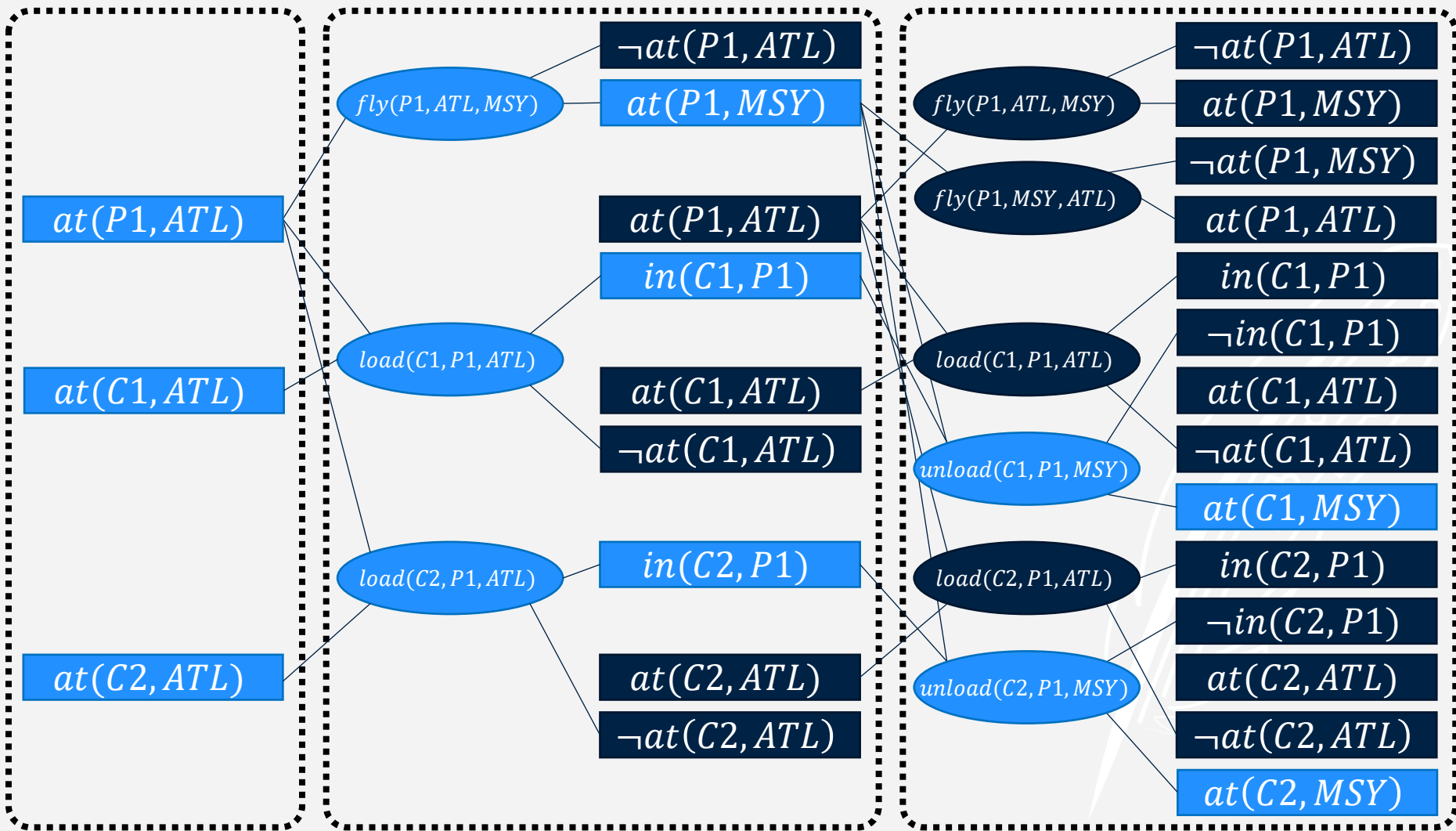
Plan Graph Estimate: ? steps

FF Estimate: ? steps

Level 0

Level 1

Level 2



Cargo Problem

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

Goal: $at(C1, MSY) \wedge at(C2, MSY)$

Plan:

1. $load(C1, P1, ATL)$
2. $load(C2, P1, ATL)$
3. $fly(P1, ATL, MSY)$
4. $unload(C1, P1, MSY)$
5. $unload(C2, P1, MSY)$

HSP Estimate: 6 steps

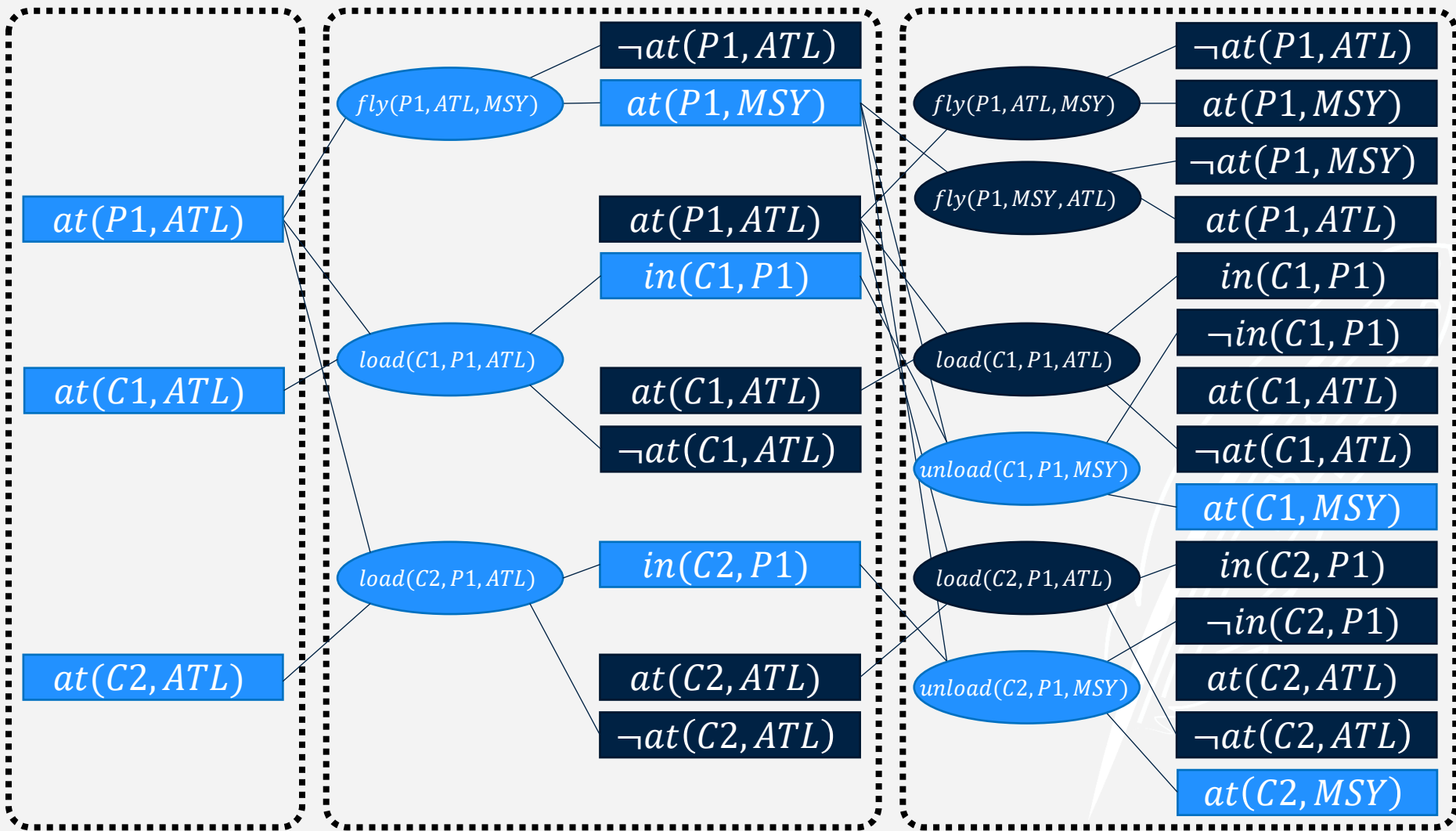
Plan Graph Estimate: 2 steps

FF Estimate: ? steps

Level 0

Level 1

Level 2



Cargo Problem

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

Goal: $at(C1, MSY) \wedge at(C2, MSY)$

Plan:

1. $load(C1, P1, ATL)$
2. $load(C2, P1, ATL)$
3. $fly(P1, ATL, MSY)$
4. $unload(C1, P1, MSY)$
5. $unload(C2, P1, MSY)$

HSP Estimate: 6 steps

Plan Graph Estimate: 2 steps

FF Estimate: 5 steps

Graphplan vs. HSP vs. FF

- All three are solving a relaxed version of the problem by never deleting facts.
- Simply using the level of the plan graph at which the goal first appears is admissible but often underestimates dramatically.
- HSP does not account for synergy between goals and so is more prone to overestimate.
- FF accounts for synergy between goals and so often gives more accurate estimates.

Mutexes

FF does not compute mutexes when extending the plan graph. This is for two reasons:

- In practice, the extra accuracy gained by using mutexes is not worth the cost of computing them.
- Graphplan solution extraction is P-SPACE-hard because of mutexes. FF must re-compute the plan graph at every iteration of the search, so extracting a solution with mutexes is way too expensive. Without mutexes, solution extraction can be done greedily and is only P-TIME-hard.

Fast-Downward

- Created by Malte Helmert and Silvia Richter
- Winner of the fourth International Planning Competition (2004)
- Every winning planner since then up until the present has been based on FD



Fast-Downward

- Translates propositional problem representation into a variable / value representation, which allows for smaller and faster data structures.
- Computes domain transition graph to describe how a variable's value can change.
- Relaxed plans extracted from domain transition graphs.

Propositional Representation

Given 1 plane and 4 airports, there are 8 literals the need to be expressed:

1. $at(P1, ATL)$
2. $\neg at(P1, ATL)$
3. $at(P1, MSY)$
4. $\neg at(P1, MSY)$
5. $at(P1, SFO)$
6. $\neg at(P1, SFO)$
7. $at(P1, DFW)$
8. $\neg at(P1, DFW)$

Traditionally, a state has been represented as an array of Boolean variables. Given a problem with p planes and a airports, how many indices are needed in this array?

$$p \cdot a$$

Variable / Value Representation

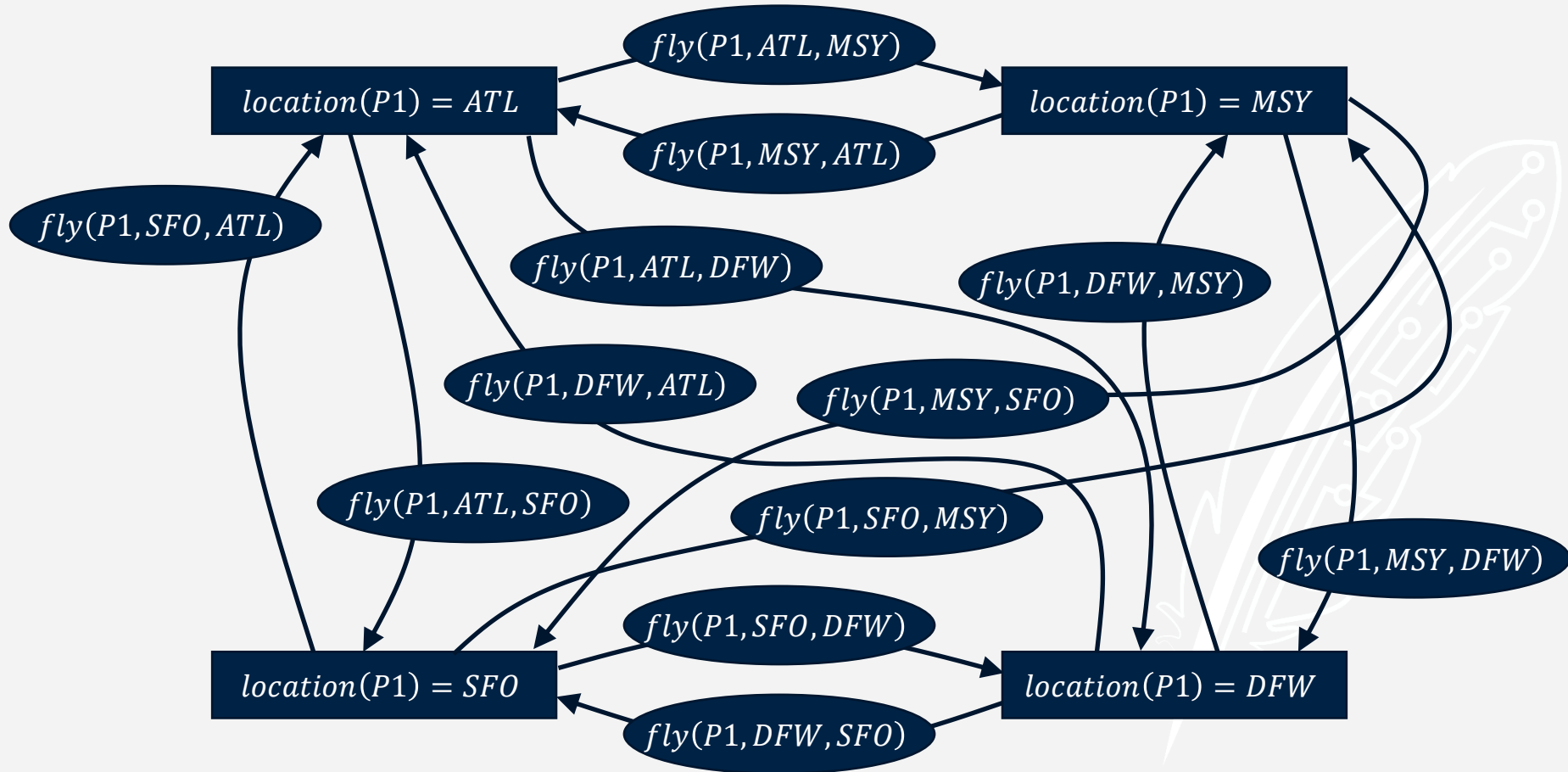
Rather than representing literals as Boolean values, FD infers a set of variables which can have one of many possible values:

1. $location(P1) = ATL$
2. $location(P1) = MSY$
3. $location(P1) = SFO$
4. $location(P1) = DFW$

Given a problem with p planes and a airports, how many indices are needed in this new array?

p

Domain Transition Graphs



FD Heuristic (Overview)

- FD calculates its heuristic by considering domain transition graphs (DTGs).
- Given the current value of a variable (current node in the DTG), follow edges (steps) until we reach the value of that variable in the goal.
- The order in which variables are considered is decided based on how goals interact.
- FD is essentially breaking down each goal into a causal chain (remember causal links?).

HSP vs. FF vs. FD

All three planners calculate their heuristics by solving a relaxed version of the problem that is only P-TIME-hard instead of P-SPACE-hard:

- HSP does not account for synergy or interference.
- FF accounts for synergy but not interference.
- FD accounts for synergy and some interference.

History of Planning

- State space planning would be the most straightforward way to approach the problem, but the search space explodes so quickly that this is only viable with highly accurate heuristics.
- Other approaches to planning (e.g. POCL and Graphplan) are developed. They use abstraction to reduce the number of decisions the planner makes.

History of Planning

- Plan graphs leads to the development of accurate heuristics.
- Heuristics become progressively more accurate by solving relaxed problems which are more and more similar to the original problem.
- Other ideas developed early in the history of planning research (e.g. causal links) are often helpful in developing new approaches.