Grounding, Search, and Heuristics in FOND HTN planning

Mohammad Yousefi

Presenter

Pascal Bercher

Supervisor

ANU/CECC

March 25, 2024

Australian National University

NATURAM PRIMUM COGNOSCERE RERUM

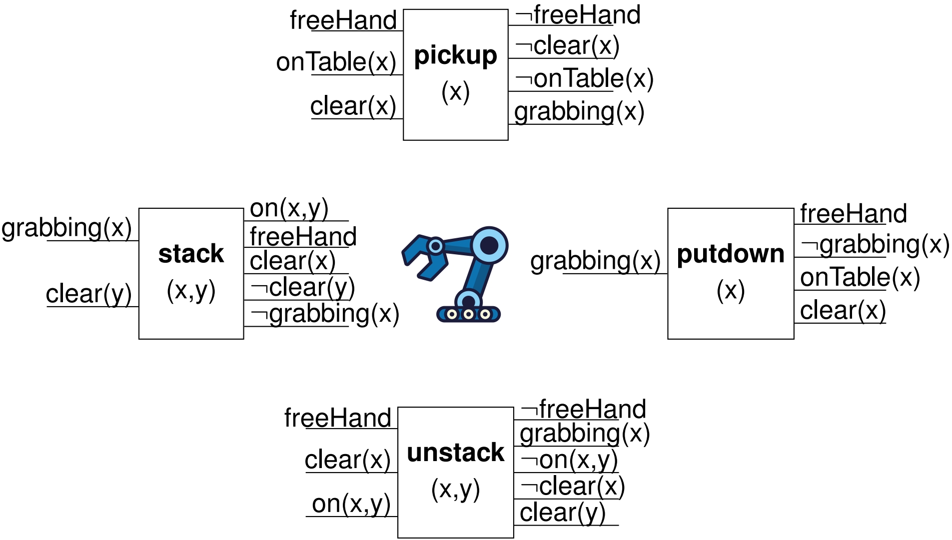
Outline

Introduction

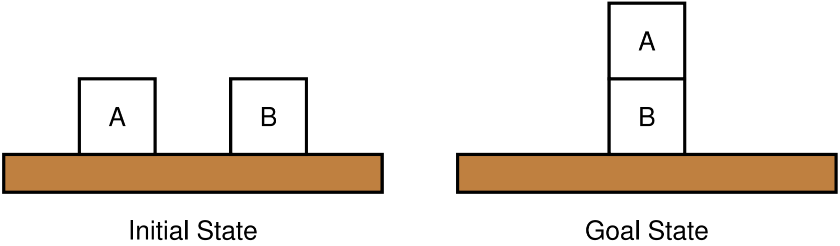
Introduction

Planning is a systematic approach to problem solving. It involves reasoning about resources, constraints, and taking actions to achieve an outcome.

Planning Domain

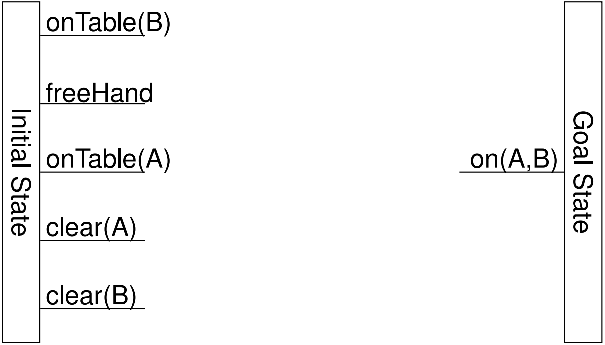


Problem Instance

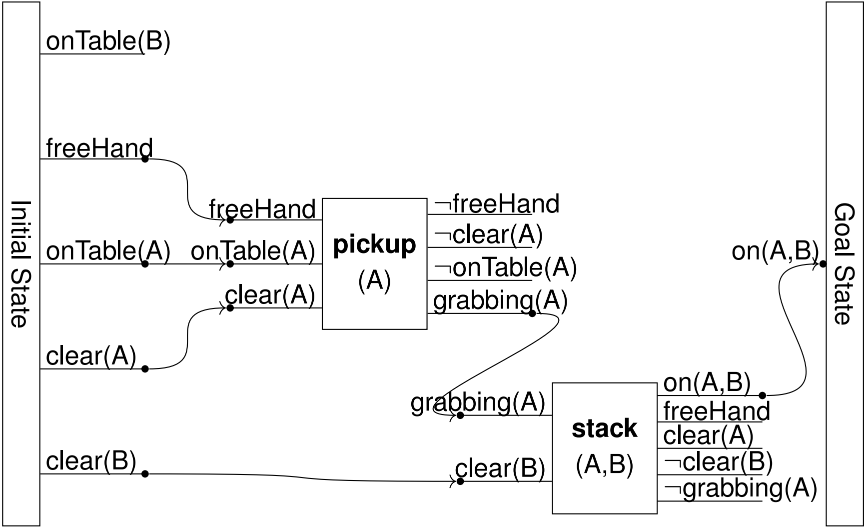


Formal Representation

Formally, a classical planning problem is a tuple $P = \langle F, A, I, G \rangle$ where F is a set of propositions, A is a set of actions, and I, G are the initial and goal states.

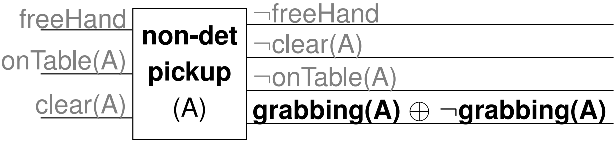


A Basic Solution



Non-Determinism

Example


freeHand
onTable(A)
clear(A)
non-det
pickup
(A)
¬freeHand
¬clear(A)
¬onTable(A)
grabbing(A) ⊕ ¬grabbing(A)

Solutions and Guarantees

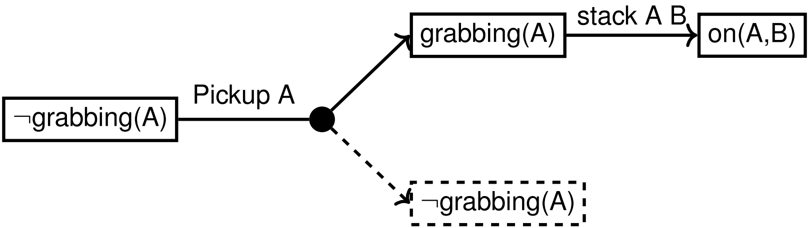
In the presence of non-determinism, in a fully-observable environment, solutions (which are called *policies*) are a mapping from observations (e.g., states) to execution steps (e.g. actions).

Solutions and Guarantees: Classical Policy

Formally, a classical FOND policy π is defined as a partial function $\pi : S \rightarrow A$ where S is the set of states, and A is the set of actions [1].

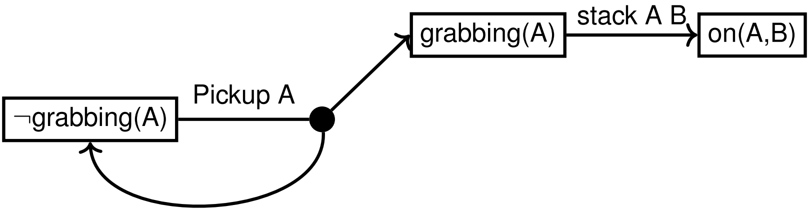
Solutions and Guarantees: Weak Solutions

Weak solutions are optimistic plans that only work if the outcomes are favourable.



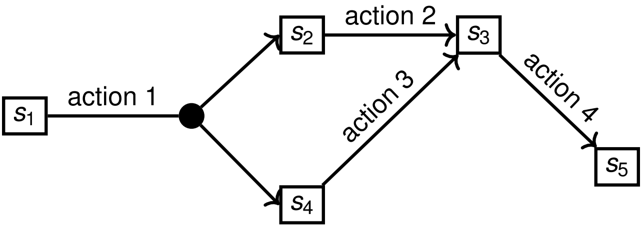
Solutions and Guarantees: Strong Cyclic Solutions

Strong cyclic solutions guarantee termination in a goal state regardless of uncertainty.



Solutions and Guarantees: Strong Solutions

Strong (acyclic) solutions guarantee termination in a goal state regardless of uncertainty, **without having a cycle.** Hence, they have a finite number of execution steps. The provided example does not have a strong solution.



Hierarchical Planning

Formalization: Motivation

In the provided example, it is easy to see that all solutions are of the form:

$$\text{[grab a block]} \rightarrow \text{[put it somewhere else]}$$

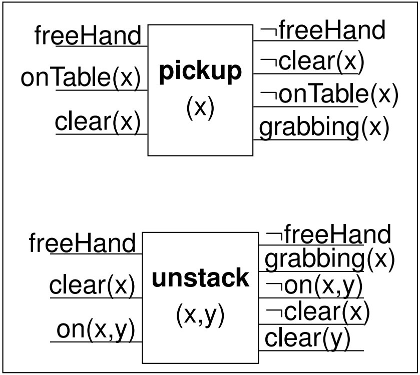
This phenomenon occurs in many planning problems, and gives us another perspective on planning, namely hierarchical refinement.

Formalization: Some More Examples

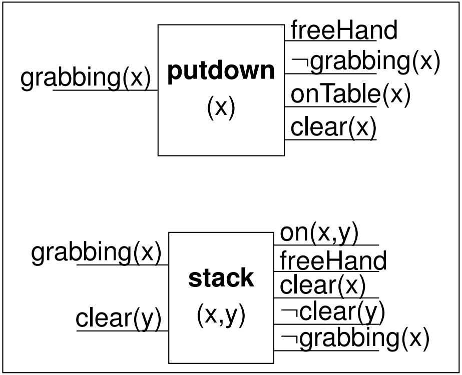
Uber Eats: [Restaurant prepares the order] $\rightarrow$ [Driver picks it up] $\rightarrow$ [Driver delivers it]
Travelling: [Go to airport] $\rightarrow$ [Fly to the destination] $\rightarrow$ [Go to Hotel]
Getting a PhD: [Do research] $\rightarrow$ [Write a dissertation]

Formalization: Compound Task

Grab(x,y)

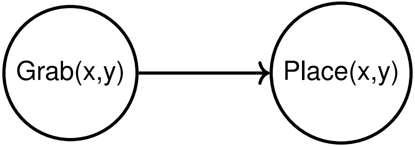


Place(x,y)



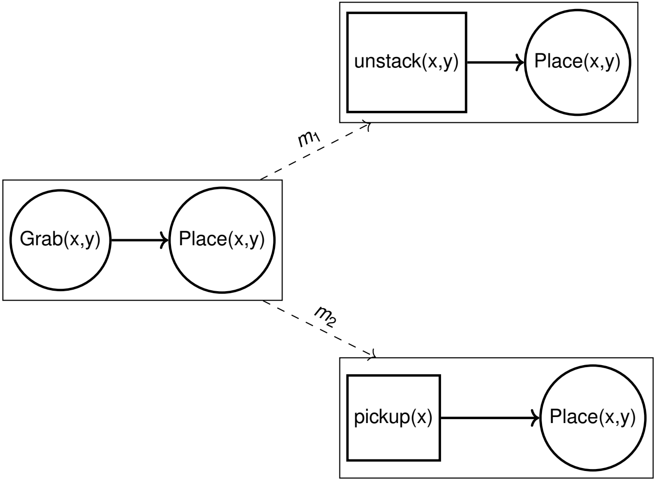
Formalization: Task Networks

A task network $tn = \langle T, \prec, \alpha \rangle$ is a partially ordered multiset of tasks.



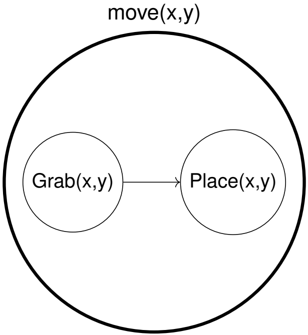
Formalization: Refinement Methods

A refinement method $m = \langle c, tn \rangle$ decomposes a compound task c to its refined task network.



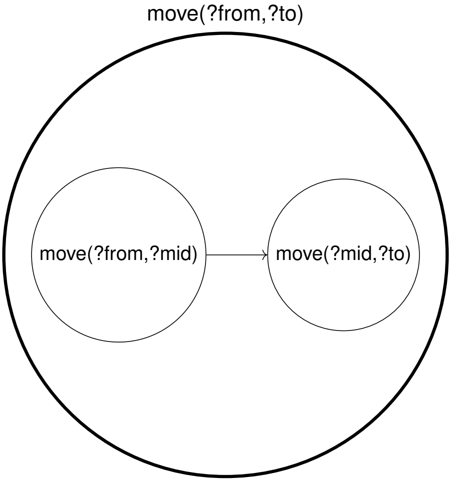
Formalization: Hierarchy

Compound tasks may contain other compound tasks as well. This allows hierarchical representation of tasks.



Formalization: Recursive Tasks

Recursion happens when a compound task contains a copy of itself.

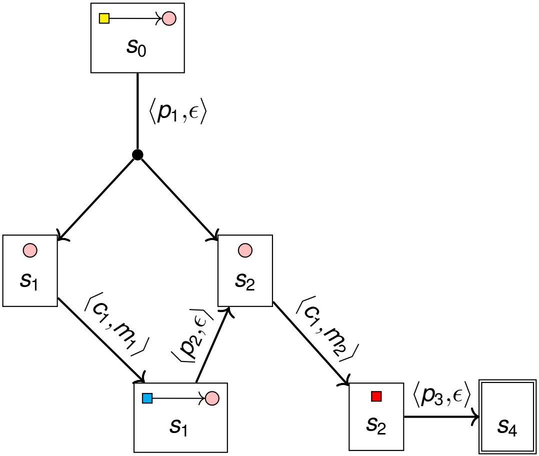


Non-Determinism in Hierarchical Planning

Two formalisms has been put forth to deal with non-determinstic tasks in HTN planning [2, 3]. We focus on the more expressive, method-based, one.

Non-Determinism in Hierarchical Planning: Method-Based Policy

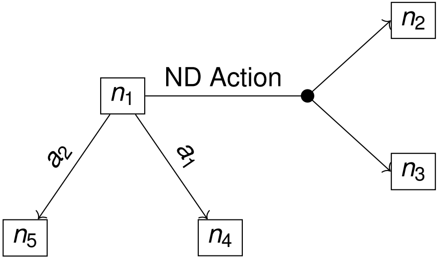
A method-based policy is a partial mapping $\pi^{MP}: \ TN \times S \rightarrow T \times M \cup \{\epsilon\}$.



AOD Relaxation

Hypergraph

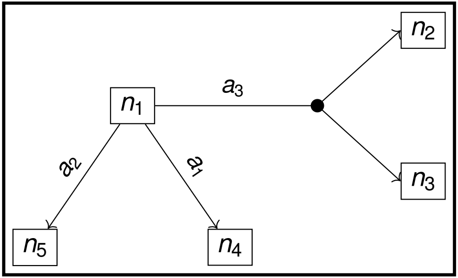
FOND problems are often represented as an AND/OR graph (also known as a hypergraph), which is a generalisation of a canonical graph that allows multiple connections using a single edge.



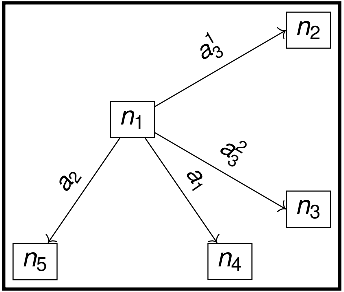
All-outcome-determinized relaxation

Our relaxation drops the “and” semantic from the search space.

Original Problem



Relaxed Problem



Grounding

What is Grounding?

Even though planners operate in grounded (i.e., variable free) domains, PDDL and HDDL allow parameterized representations. For example `unstack(?x, ?y -block)` is a generic action applicable to all `?x` and `?y` of type block.

Our Approach

Given a set of non-deterministic action names, N_{nd}, and a valid grounding obtained for the AOD-problem:

1. For all ground compound tasks $C[o_1, o_2, ..., o_k]$, if $C \in N_{nd}$ and it has as many methods as the number its outcome, infer its grounding $[o_1, o_2, ..., o_k]$ as a valid grounding for the corresponding non-deterministic action, otherwise skip.
2. Copy all remaining groundings (those not covered in the previous loop, i.e., actions), to the valid groundings in the non-deterministic domain.

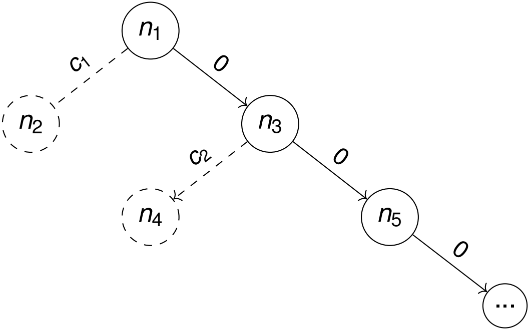
Heuristic Search

Search: Non-Recursive Domains

- **Weak Solutions**: A^*
- **Strong Solutions**:
  - Acyclic Search Spaces: $AO^*(CF)$
  - Cyclic Search Spaces: $AO^*(CFC\_Rev^*)$
- **Strong Cyclic Solutions**: LAO^*

Search: Recursive Domains

Recursive domains make search **very difficult** because they introduce infinite-length zero-cost paths.



Search: Recursive Domains

The solution to this problem is still an open research question for us, but possible solution candidates are:

- “pseudo” multi-objective search, and
- randomized tie-breaking.

Heuristic: Definition

Heuristics are easy-to-compute information that guides the search process toward goal nodes. They are usually obtained from a relaxed version of the problem.

Heuristic: Our Approach

We have relaxed the undecidable HTN planning problem to a polynomial time computable heuristic function through a cascade of relaxations.

1. Relax the FOND HTN problem (undecidable) to a deterministic one (undecidable) using the all-outcome-determinization procedure.
2. Relax the deterministic problem to a classical planning problem (PSPACE-Complete) using the Relaxed Composition (RC) procedure.
3. Use any deterministic classical heuristic e.g., FF (Poly-Time), and Add (Poly-Time).

Result

Setup

- We have modified 5 hierarchical domains of IPC '20 to include non-deterministic actions,
- Each domain has 15 problem instances,
- 30 min cut-off time.

Domain	Problems	IPC Score		Coverage		Avg. Execution Structure			
						# of Nodes		CP Length	
		Add	FF	Add	FF	Add	FF	Add	FF
Depots	15	6.92	6.99	10	10	150.80	150.80	127.20	127.90
Rover	15	5.16	5.00	6	5	28.00	23.80	21.83	17.80
Snake	15	2.54	2.45	4	4	32.75	32.75	27.00	27.00
Satellite	15	5.92	4.16	7	7	33.86	33.71	16.86	16.86
Transport	15	3.48	4.04	5	5	400.60	99.20	45.40	33.80

Bibliography

[1] A. Cimatti, M. Pistore, M. Roveri, and P. Traverso. “Weak, strong, and strong cyclic planning via symbolic model checking”. In: *Artificial Intelligence* 147.1 (2003), pp. 35–84.

[2] D. Chen and P. Bercher. “Fully Observable Nondeterministic HTN Planning – Formalisation and Complexity Results”. In: *Proceedings of the 31st International Conference on International Conference on Automated Planning and Scheduling*. AAAI Press, 2021, pp. 74–84.

[3] D. Z. Chen and P. Bercher. “Flexible FOND HTN Planning: A Complexity Analysis”. In: *Proceedings of the 32nd International Conference on International Conference on Automated Planning and Scheduling*. AAAI Press, 2022, pp. 26–34.

THANK YOU!