Towards Search Node-Specific Special-Case Heuristics for HTN Planning – An Empirical Analysis of Search Space Properties under Progression

Lijia Yuan and Pascal Bercher

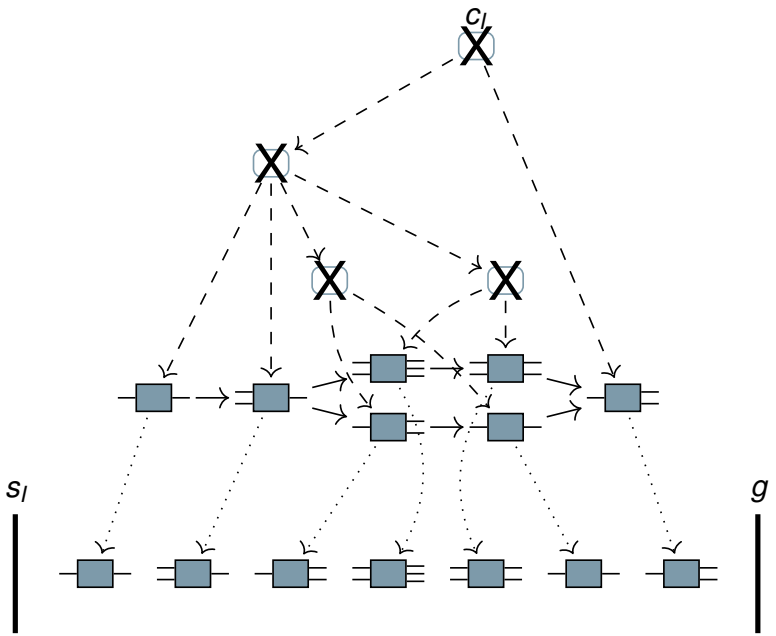
School of Computing
College of Engineering, Computing and Cybernetics
The Australian National University

3 June 2024

Australian National University

Hierarchical Task Network Planning

Introduction to HTN Planning



$$\mathcal{P} = (V, P, \delta, C, M, s_I, c_I, g)$$

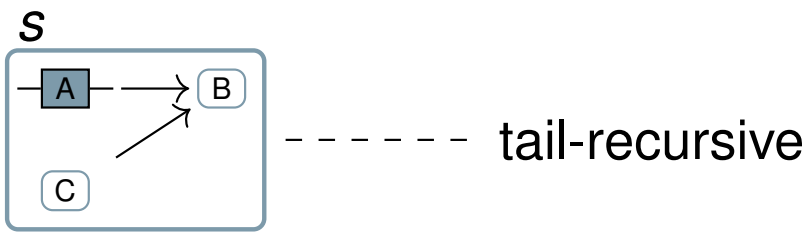
- V a set of state variables
- P a set of primitive task names
- $\delta : P \rightarrow (2^V)^3$ the task name mapping
- C a set of compound task names
- $c_I \in C$ the initial task
- $M \subseteq C \times 2^{TN}$ the methods
- $s_I \in 2^V$ the initial state
- $g \subseteq V$ the (optional) goal description

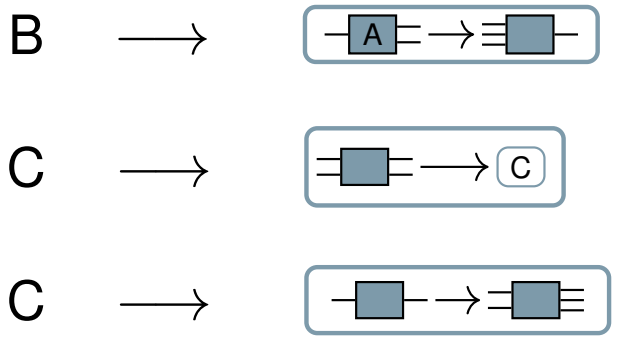
A solution task network *tn* must:

- be a refinement of c_I,
- only contain primitive tasks, and
- have an executable linearization that makes the goals in g true.

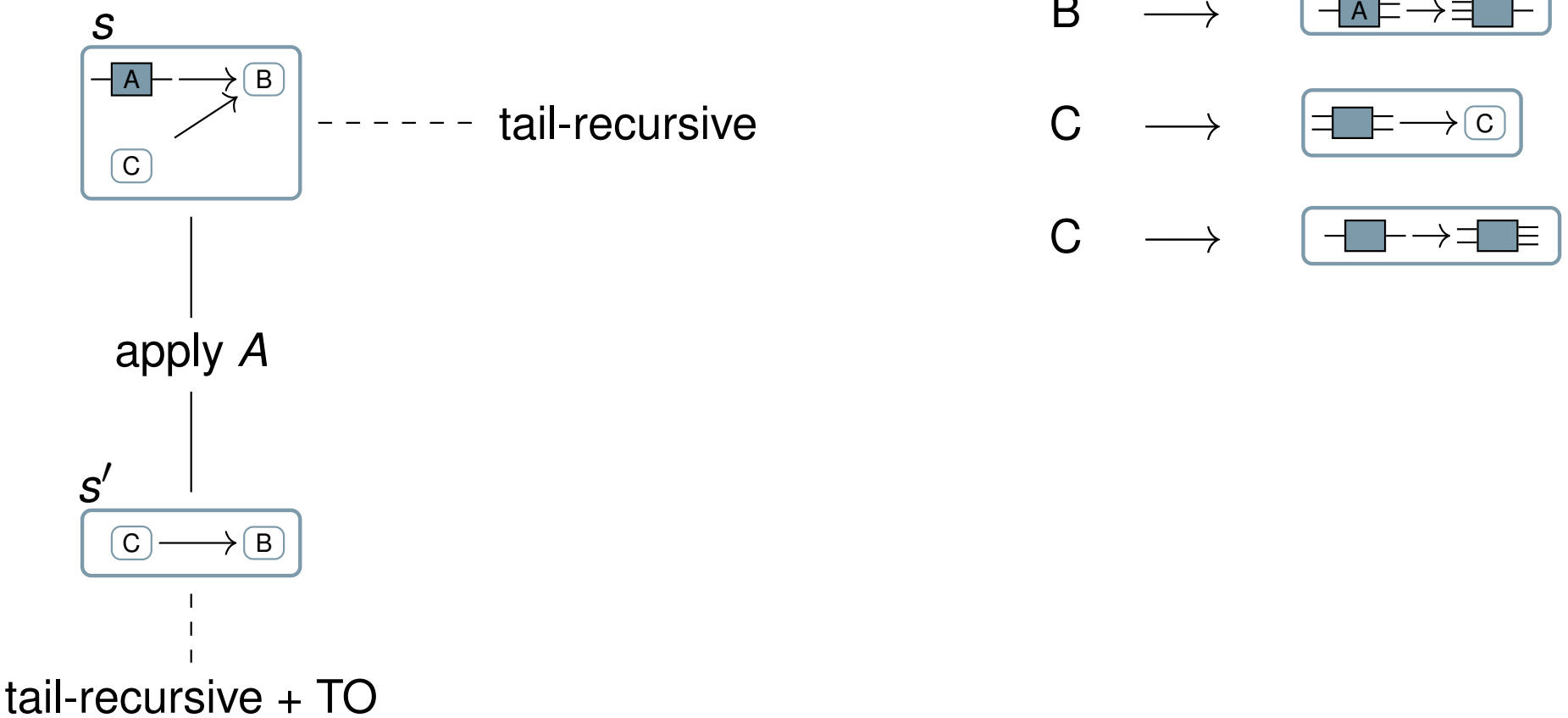
Special Cases during Progression Search

We can do better! (Make search systematic)

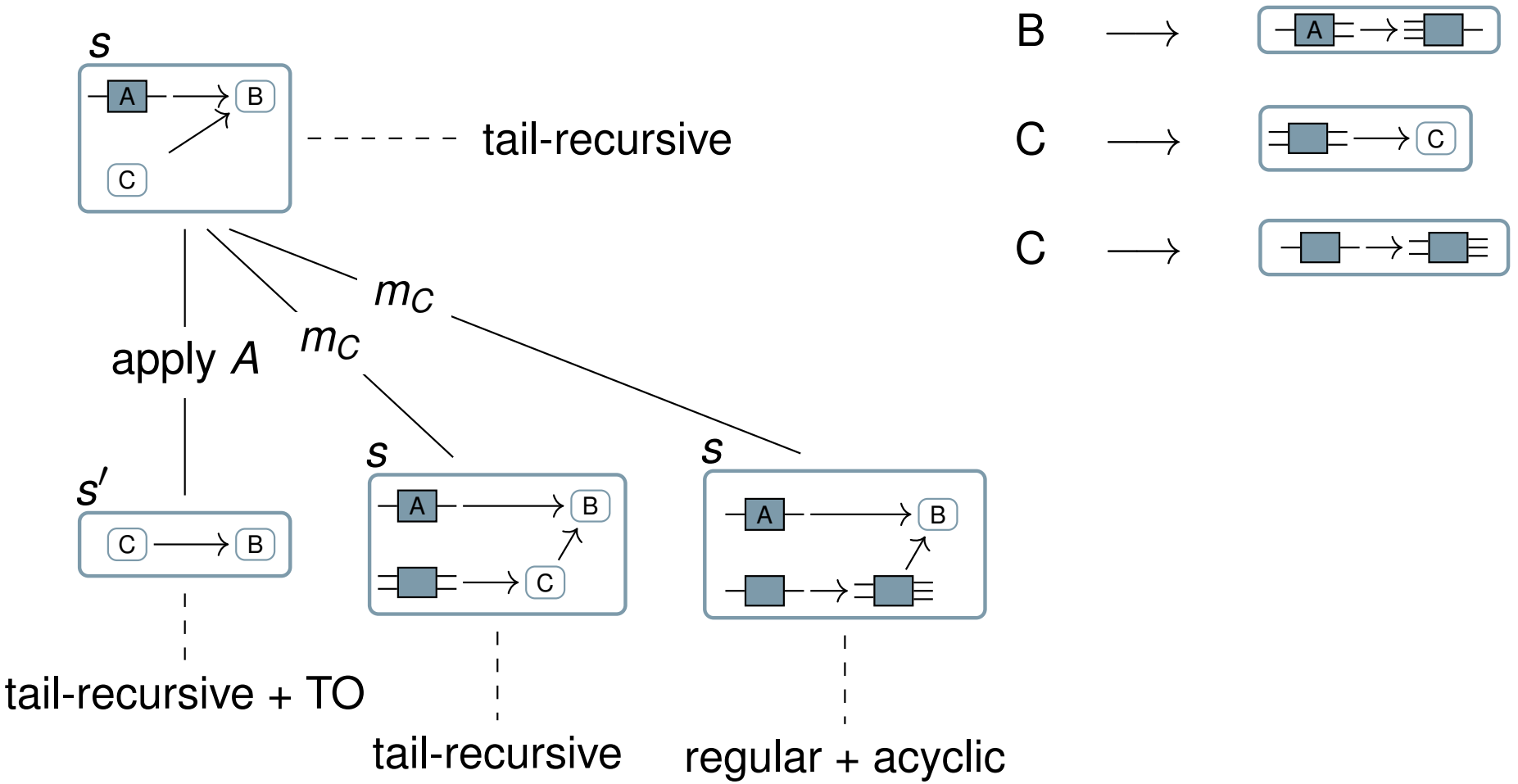




We can do better! (Make search systematic)



We can do better! (Make search systematic)



Our Motivation and Investigations

Results

Motivation

- We can deploy – *dependent on search nodes* – specialized:
  - heuristics
  - any techniques, like compilations, pruning, etc.

Results

Motivation

- We can deploy – *dependent on search nodes* – specialized:
  - heuristics
  - any techniques, like compilations, pruning, etc.

Our work:

- Measure problem class frequency: Is detection effort worth it?!
- Propose two techniques for detection:
  - one cheap, but does not update impact of action progressions
  - one more expensive, and obtains tighter problem classes