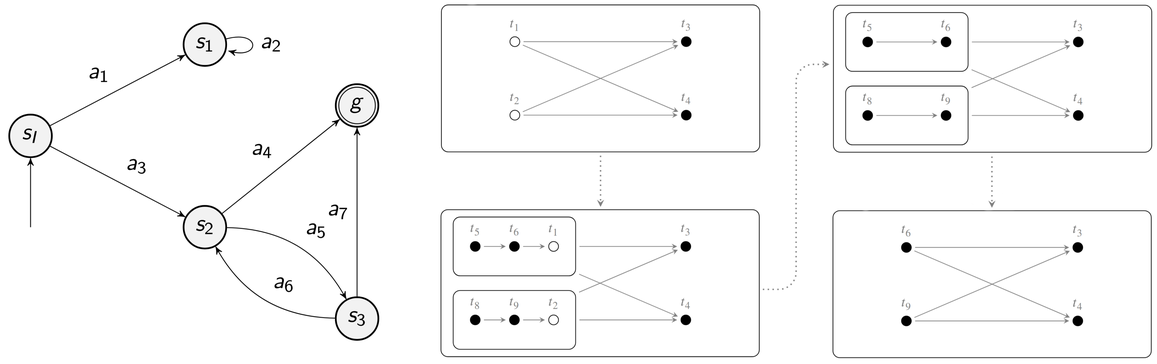
# Fully Observable Nondeterministic HTN Planning – Formalisation and Complexity Results

**Dillon Chen, Pascal Bercher**
The Australian National University
{dillon.chen, pascal.bercher} @anu.edu.au

## HTN planning in a nutshell

- Aim of classical planning: reach a goal state with a sequence of actions.
- Aim of HTN planning: execute a given set of tasks with task decomposition.
  - tasks either compound or primitive
  - compound tasks can be decomposed into another set of tasks
  - primitive tasks = actions




## Why HTN planning?

- Expressive - complexity ranges from tractable to undecidable
- Nice compilation from classical planning
- Easy to encode domain dependent knowledge
- Levels of abstraction helpful for communicating with users

## Adding uncertainty

- Classical planning: actions may have several effects
- HTN planning: actions may have several effects
  - same!

## Defining solutions

- Nondeterministic planning: sequence or policy of actions
- Nondeterministic HTN planning: decompose away all compound tasks followed by sequence or policy of actions
  - alternatively, integrate decomposition into policy

executing a policy $\pi$:

- $s \leftarrow s_I$
- while $\pi(s)$ exists:
  - execute $\pi(s)$
  - $s \leftarrow$ senseState()
  - if $s = s_G$: return success

## Complexity results

- (almost all) weak FOND HTN problems can be compiled into deterministic problems
- totally ordered FOND HTN problems can be compiled into deterministic problems
- partially ordered FOND HTN problems made at least one class harder

| Hierarchy | Order | FOD | FOND: Weak | | FOND: Strong: *linearisation-dependent* | | FOND: Strong: *outcome-dependent* | |
|---|---|---|---|---|---|---|---|---|
| primitive | total | P* | NP | [4.1] | P* | [4.8] | | |
| | partial | NP$^{\alpha}$ | NP | [4.2] | NP | [4.7] | PSPACE | [5.1] |
| no recursion (acyclic) | total | PSPACE$^{\beta}$ | PSPACE | [4.4] | PSPACE | [4.8] | | |
| | partial | NEXPTIME$^{\beta}$ | NEXPTIME | [4.4] | NEXPTIME | [4.7] | EXPSPACE* | [5.2] |
| regular | total | PSPACE$^{\alpha}$ | PSPACE | [4.5] | PSPACE | [4.8] | | |
| | partial | PSPACE$^{\alpha}$ | PSPACE | [4.5] | PSPACE | [4.7] | EXPSPACE* | [5.3] |
| tail-recursion | total | PSPACE$^{\beta}$ | PSPACE | [4.4] | PSPACE | [4.8] | | |
| | partial | EXPSPACE$^{\alpha,\beta}$ | EXPSPACE | [4.4] | EXPSPACE | [4.7] | semidecidable* | [3.1] |
| arbitrary recursion | total | EXPTIME$^{\beta}$ | EXPTIME | [4.4] | EXPTIME | [4.8] | | |
| | partial | semi- & undecidable$^{\alpha,\gamma}$ | semi- & undecidable | [3.1] | semi- & undecidable | [3.1] | semi- & undecidable | [3.1] |

International Conference on Automated Planning and Scheduling (ICAPS 2021)