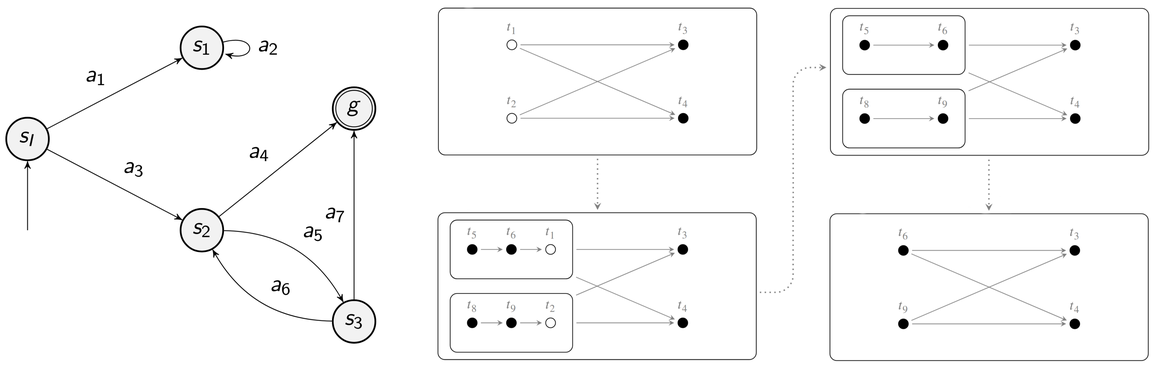
The Complexity of Flexible FOND HTN Planning

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: policy of actions
- Nondeterministic HTN planning: policy of *instructions*
  - instructions = decomposition of compound tasks or execution of primitive tasks

executing a policy π:

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

executing an HTN policy π:

- $(s, tn) \leftarrow (s_I, tn_I)$
- while $\pi(s)$ exists:
  - perform $\pi(s)$
  - $s \leftarrow$senseState()
  - $tn \leftarrow$updateTaskNetwork()
  - if $tn = \emptyset$: return success

Complexity results

- problems are all made one class harder with nondeterminism
- nonhierarchical planning analogy: PSPACE-complete to EXPTIME-complete
- idea: progression algorithms are generalised using ATMs/AND-OR trees:
  - SPACE$(f(n))$ = $\bigcup_{c>0}$ATIME$(c^{f(n)})$
  - TIME$(f(n))$ = ASPACE$(f(n))$

Hierarchy	Order	Classical/Weak		Strong		Strong cyclic	
primitive	total	P*/NP	[4.3]			P*	[4.2]
	partial	NP	[4.3]			PSPACE	[4.2]
acyclic	total	PSPACE	[4.3]			EXPTIME	[4.4]
	partial	NEXPTIME	[4.3]			EXPSPACE	[4.5]
regular	total	PSPACE	[4.3]	EXPTIME	[4.6]	EXPTIME	[4.6]
	partial	PSPACE	[4.3]	EXPTIME	[4.6]	EXPTIME	[4.6]
tail-recursive	total	PSPACE	[4.3]	EXPTIME	[4.7]	EXPTIME	[4.7]
	partial	EXPSPACE	[4.3]	2-EXPTIME	[4.8]	2-EXPTIME*	[4.8]
arbitrary	total	EXPTIME	[4.3]	semi-decidable*		semi-decidable*	
	partial	semi-/undecidable		semi-/undecidable		semi-/undecidable	