# Block Compression and Invariant Pruning for SAT-based TO-HTN Planning

Gregor Behnke (behnkeg@informatik.uni-freiburg.de)

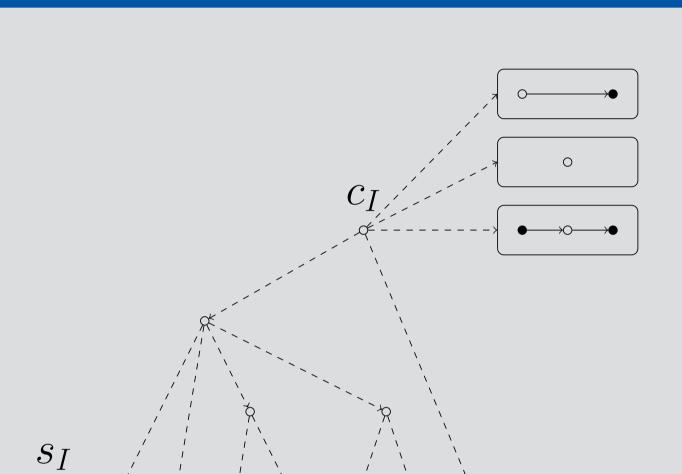
University of Freiburg



SAT-based TO-HTN planners have proven quite efficient.

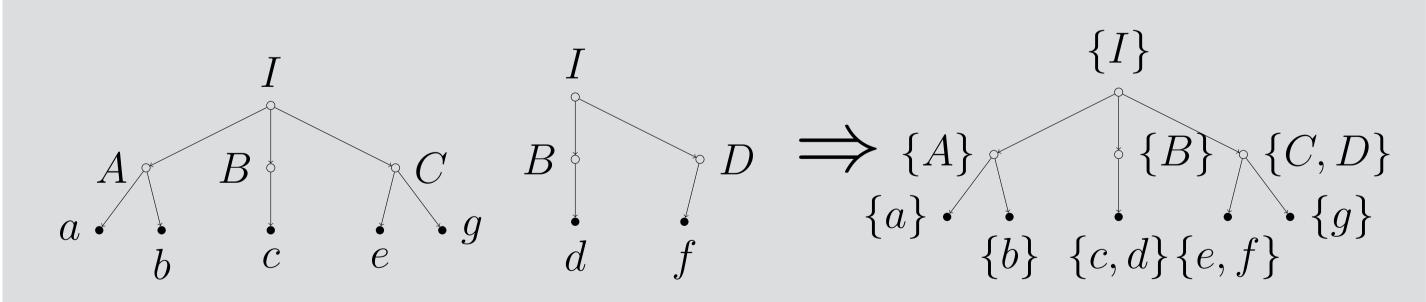
But they still encode useless actions and no parallelism!

# HTN Planning

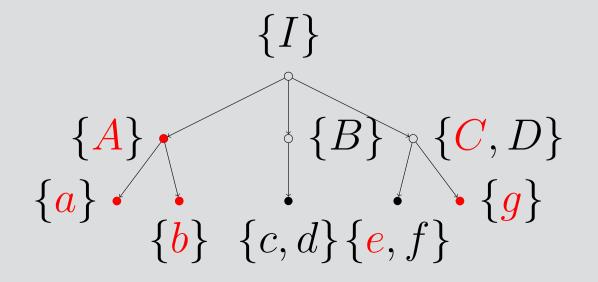


- ► Abstract tasks *A*
- ► Primitive actions *P*
- ► Decomposition methods *M*
- Preconditions / Effects for actions
- ightharpoonup Initial state  $s_I$  and initial task  $c_I$
- Note: Find a refinement  $\pi = \langle p_1, \dots, p_n \rangle$  of  $c_I$  containing only primitive actions that is **executable** in  $s_I$ .

### SAT Encoding and PDTs

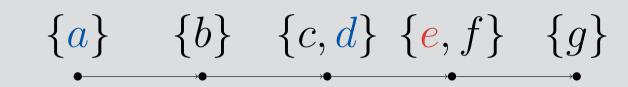


#### Pruning Propagation



- ► Assume: *a* cannot be executed
- ► Thus the method decomposing *A* becomes impossible
- ► Thus *b* is impossible
- ightharpoonup Thus C, g and e are impossible

#### Leaf Pruning



- ► Ideally: Is action part of any plan?  $\Rightarrow \mathbb{E}\mathbb{XPTIME}$ -complete
- ► Approximation: For an action *a*, can we select actions from preceding leafs s.t. *a* is executable?
  - $\Rightarrow \mathbb{NP}$ -complete

- ➤ Option 1: Delete-Relaxation
  - ► Maintain reachable facts *F*
  - ▶ Prune if  $F \not\models pre(a)$ , else add add(a) to F
- ► Option 2: Binary Invariants

  - ► Prune if  $I \not\models pre(a)$ , else update I [Rintanen'98]

## Block Compression

 $\{a\} \qquad \{b\} \qquad \{c,d\} \quad \{e,f\} \qquad \{g\}$ 

- ► Usually, each leaf is encoded as a separate time step
- ► Actions might be independent!
- ⇒ Unify time steps

All actions a must be

- ▶ non-dependent on previous actions a'  $prec(a) \cap (add(a') \cup del(a')) = \emptyset$
- non-interfering with previous actions a'  $add(a) \cap del(a') = \emptyset$   $del(a) \cap add(a') = \emptyset$

#### Evaluation on the IPC 2020 Benchmark Set

