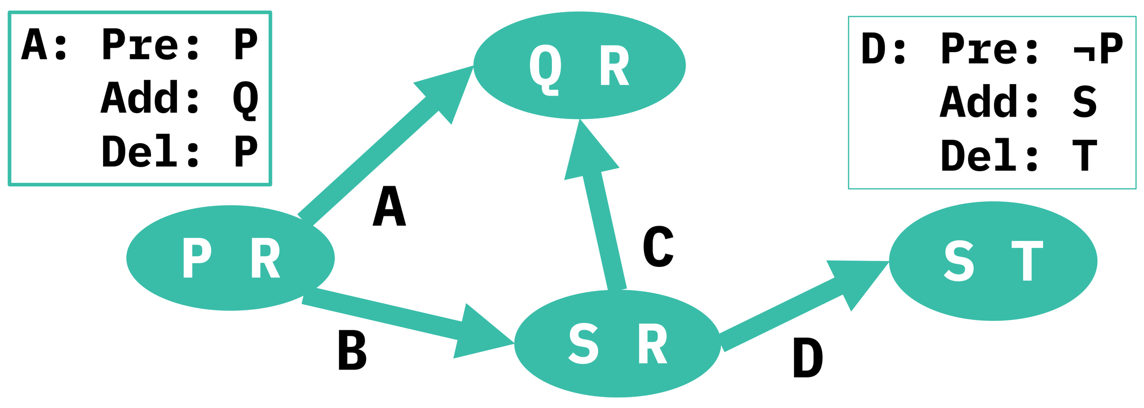
Action Space Reduction for Planning Domains

Harsha Kokel, Junkyu Lee, Michael Katz, Kavitha Srinivas, Shirin Sohrabi

The action space resulting from lifted action models in classical planning tasks is vast, with numerous redundant actions. Our research introduces an automated approach to diminish the action space by identifying seed parameters of the lifted action models.

Motivation



- Planning tasks, as labeled transition systems, use a unique label for each ground action
- Such labeling strategy generates a huge action space; which is especially unfitting for model-free RL approaches
- Above, the precond. of actions A & D are mutually exclusive, they both can never be applied in the same state. Can we use same label for them?
- We investigate if this granularity of assigning unique labels is essential or can we assign the same labels to two different ground actions and obtain a smaller action space?

Key Insight

- Action labels are used to distinguish *applicable* actions in a state.
- Providing the same label to two ground actions with mutually exclusive precond. can produce a smaller action space.
- Different assignments to some parameters of the lifted action model are guaranteed to be mutually exclusive. So, ignoring them while labeling reduces the action space.

Definitions

- *A label reduction function $\psi: L \mapsto L'$ is valid if any two distinct ground action labels $head(o_1), head(o_2) \in L$ that are applicable in the same reachable state ($s \vDash pre(o_1) \wedge s \vDash pre(o_2)$) are assigned distinct labels, that is $\psi(head(o_1)) \neq \psi(head(o_2))$.*
- *A set of ground actions $\mathcal{O}'$ is an applicable action mutex group (AAMG) if for any reachable state s,*
  $$|\{o \mid s \vDash \text{pre}(o), o \in \mathcal{O}'\}| \leq 1$$
- *An invariant candidate is a tuple consisting* ⟨fixed variable, counted variable, atom⟩ where, different groundings of fixed variables generate different sets of ground atoms and different grounding of counted variable generates ground atoms within each set.
- *An invariant candidate is called a lifted mutex group (LMG) if all of its ground atom sets are mutex groups.*

Parameter Seed Set

TLDR:
Some parameters help distinguish applicable action, some do not. The ones that do are called parameter seeds. Identification of parameter seeds can provide valid label reduction.

Th. 1: Given a lifted action o and a lifted mutex group $l = \langle v^f, v^c, \{\alpha\} \rangle$, if p $\sqsubseteq \alpha$ for some p $\in$ pre(o) , then any assignment c to $X = \text{params}(o) \setminus v^c(l)$ results in $o_{\downarrow}(X/c)$ being an AAMG.

- Relevant LMG Conditions
  - atom of LMG is part of precondition
  - variable types in LMG is a super-type of the variable of the action parameter type
- LMG guarantees unique assignment to $v^c(l)$ in any given state, so $v^c(l)$ can be removed from the parameter set while assigning labels. This reduces the action space.
- Multiple LMGs can be further used to remove other non-essential parameters.

Parameter seed set problem

Input: A lifted action o with parameters params(o) and a set of relevant lifted mutex groups L.
Find: A subset $X \subseteq$ params(o) of parameters s.t. $\exists X_1, \ldots X_k$ with
(i) $X = X_1 \subseteq X_2 \subseteq \cdots \subseteq X_k = \text{params}(o)$, and
(ii) $X_{i+1} = X_i \cup v^c(l)$ for some $l \in L$ s.t. $v^f(l) \subseteq X_i$.

Parameter seed set identification

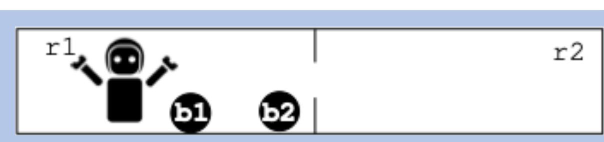
1. Find *relevant* Lifted Mutex Groups (LMG)
2. Define following delete-free planning problem
   $\Pi_o = \langle \mathcal{L}_o, \mathcal{O}_o, I_o, G_o \rangle$, where
   - Language $\mathcal{L}_o$ contains a single predicate mark and an object for each parameter in $params(o)$.
   - The set $\mathcal{O}_o$ consists of two types of actions
     1. seed$_x$ actions are defined for each parameter $x \in params(o)$ as seed$_x$:= $\langle \text{seed}_x, log(|\mathcal{D}(x)|), \emptyset, \{\text{mark}(x)\}, \emptyset \rangle$
     2. get$_l$ actions are defined for each relevant LMG l as get$_l$:= $\langle get_l, 0, \{\text{mark}(x) \mid x \in v^f(l)\}, \{\text{mark}(y) \mid y \in v^c(l)\}, \emptyset \rangle$.
   - Initial state $I_o = \emptyset$
   - Goal state $G_o = \{\text{mark}(x) \mid \forall x \in params(o)\}$.
3. Find a plan π for Π_o,
   $X_\pi = \{c \mid \text{seed}_c \in \pi\}$ is a set of parameter seeds

Proof of correctness

Th. 2: Let o be a lifted action over parameters params (o) and X be a solution to the parameter seed set problem above. Any assignment c of objects to X results in $o_{\downarrow}(X/c)$ being an AAMG.

Th. 3: For a plan π of $\Pi_o, X_\pi = \{c \mid \text{seed}_c \in \pi\}$, is a solution to the parameter seed set problem of o.

Example




```
(:action pick
 :parameters(?b ball, ?r room, ?g gripper)
 :precondition(and (at ?b ?r) (at-robby ?r)
                   (free ?g))
 :effect (and(not(at ?b ?r))(not (free ?g))
                (carry ?b ?g)))
```

- In gripper domain
  `(pick b1 r1 g1)` ⊕ `(pick b1 r2 g1)`
  because of mutually exclusive preconditions
  `(at b1 r1)` ⊕ `(at b1 r2)`

Lifted Mutex Group:

```
l = ⟨{?b},{?r},{at(?b,?r)}⟩
l↓(?b/b1)={at(b1,r1), at(b1,r2)}
l↓(?b/b2)={at(b2,r1), at(b2,r2)}
```

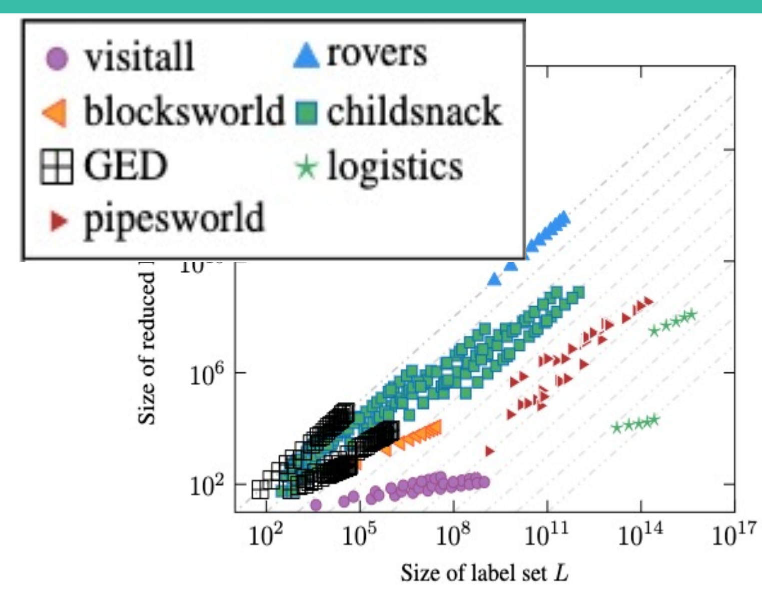
Applicable action mutex group:

```
(pick b1 r1 g1),
(pick b1 r2 g1),
(pick b1 r3 g1),
 . . .
```

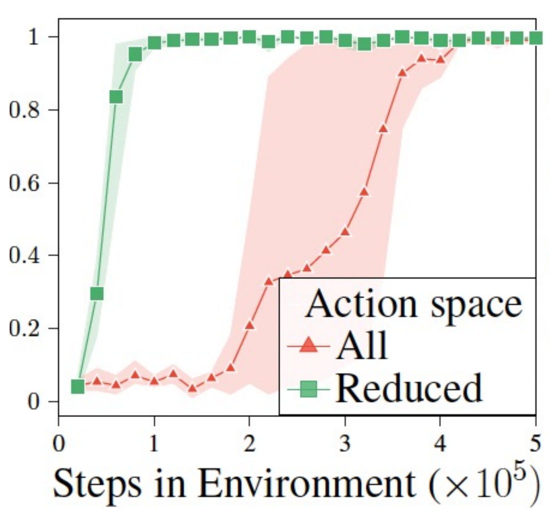
```
(pick b2 r1 g1),
(pick b2 r2 g1),
(pick b2 r3 g1),
 . . .
```

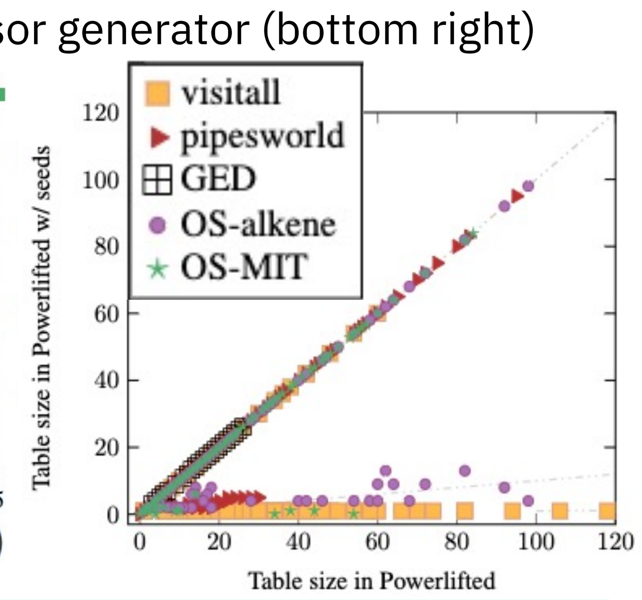
So, use label
`(pick ?b ball, ?g gripper)`
instead of
`(pick ?b ball, ?r room, ?g gripper )`

Results



- **Significant reduction in action spaces**; going up to 2 orders of magnitude on IPC domain problems and up to 10 orders of magnitude on hard-to-ground domain problems (above)
- **Sample efficient for learning RL agents**; reduces sample requirements in the Gripper domain (bottom left) and makes RL training feasible on blocks domain
- **Reduction in table sizes** for applicable action queries in lifted successor generator (bottom right)



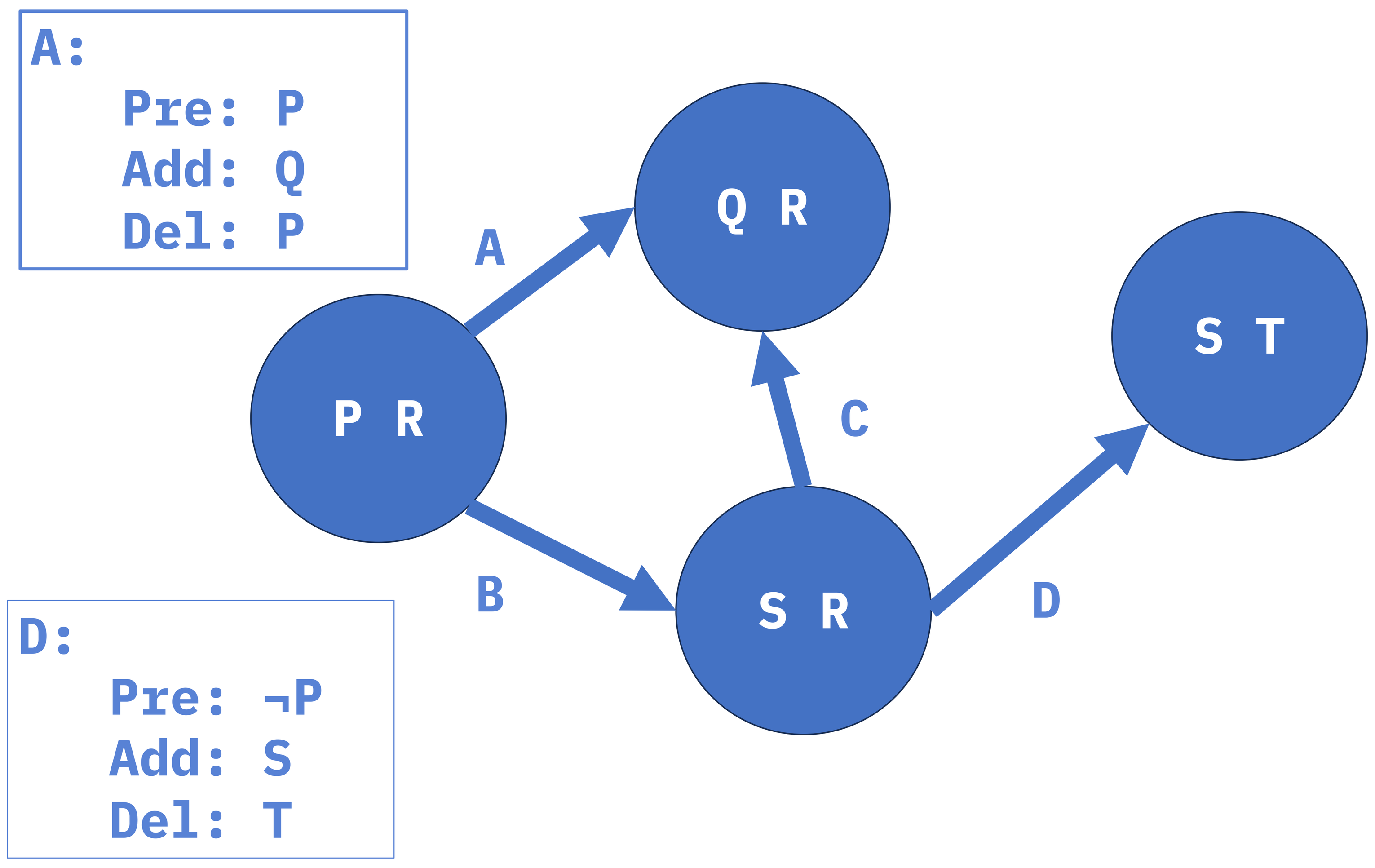


- **Contributions:**
  1. Characterize a valid label reduction
  2. Propose an automatic *valid* label reduction approach
     - Demonstrate that our approach achieves a significant reduction
  3. Two use-cases
     - Model-free reinforcement learning
     - Lifted successor generation

IBM **Research**

</> https://ibm.github.io/Parameter-Seed-Set/

hk harsha.kokel@ibm.com


A:
Pre: P
Add: Q
Del: P
A
Q R
P R
S T
C
B
S R
D
D:
Pre: ¬P
Add: S
Del: T