Action Space Reduction for Planning Domains

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

IBM Research

Actions as labels

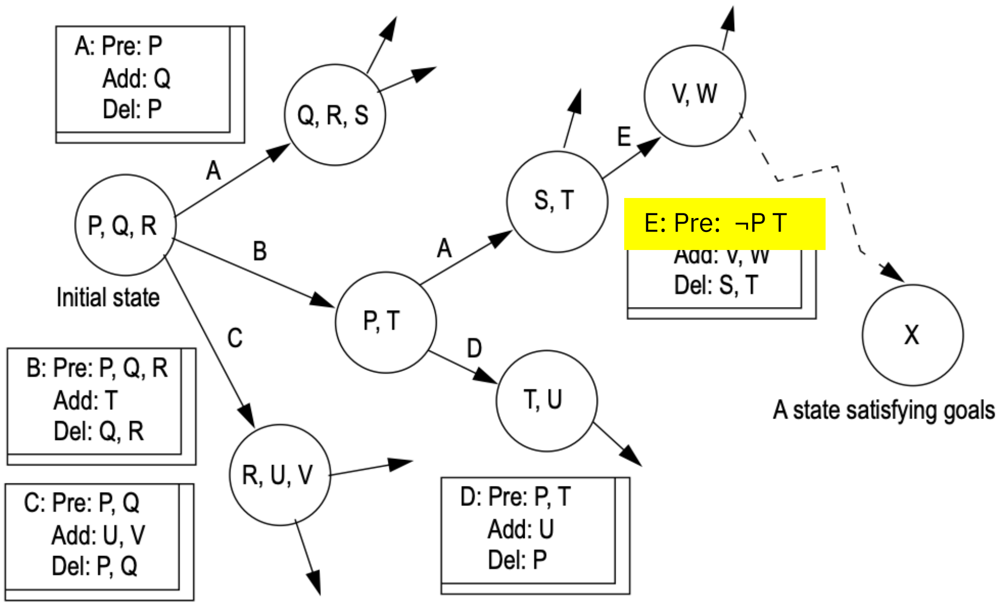
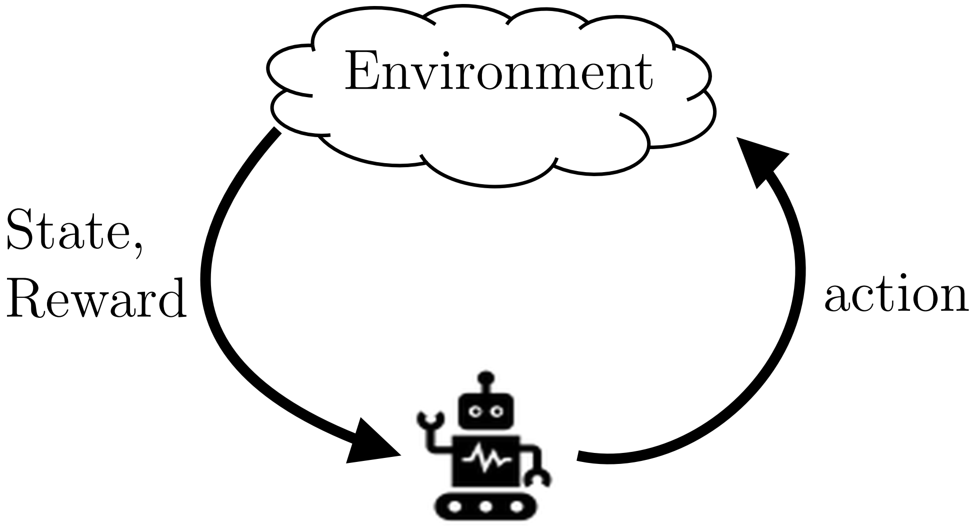


Image source: Long and Fox (2002)

Reinforcement Learning



Markov Decision Process
(MDP)

$\langle S, A, T, R \rangle$

S: State
A: Action labels / space
T: Transition function
R: Reward function

Contributions

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

Valid Label Reduction

A label reduction function $\psi: L \mapsto L'$ is *valid* if any two distinct ground action labels $\mathrm{head}(o_1), \mathrm{head}(o_2) \in L$ that are applicable in the same reachable state $\left(s \vDash \mathrm{pre}(o_1) \ \wedge \ s \vDash \mathrm{pre}(o_2)\right)$ are assigned distinct labels, that is $\psi\left(head(o_1)\right) \neq \psi\left(head(o_2)\right)$.

Gripper domain

Gripper task $\Pi = \langle \mathcal{L}, \mathcal{O}, I, G \rangle$.

- Language $\mathcal{L}$ includes:
  - objects $\mathcal{B}$: $\mathtt{r1}, \mathtt{r2}, \mathtt{b1}, \mathtt{b2}, \mathtt{g1}, \mathtt{g2}$
  - types $\mathcal{T}$: $\mathtt{room}, \mathtt{ball}, \mathtt{gripper}$
  - variables $\mathcal{V}$: $?\mathtt{r}, ?\mathtt{b}, ?\mathtt{g}, ?\mathtt{f}, ?\mathtt{t}$
  - predicates $\mathcal{P}$: $\mathtt{at\_robby}, \mathtt{at}, \mathtt{free}, \mathtt{carry}$

- Schematic operators $\mathcal{O}$ consists of:

$\mathtt{move}$: params $\{?\mathtt{f} : \mathtt{room},\ ?\mathtt{t} : \mathtt{room}\}$
  : pre $\{\mathtt{at\_robby}(?\mathtt{f})\}$
  : add $\{\mathtt{at\_robby}(?\mathtt{t})\}$
  : del $\{\mathtt{at\_robby}(?\mathtt{f})\}$

$\mathtt{pick}$: params $\{?\mathtt{b} : \mathtt{ball}, ?\mathtt{r} : \mathtt{room}, ?\mathtt{g} : \mathtt{gripper}\}$
  : pre $\{\mathtt{at}(?\mathtt{b}, ?\mathtt{r}),\ \mathtt{at\_robby}(?\mathtt{r}), \mathtt{free}(?\mathtt{g})\}$
  : add $\{\mathtt{carry}(?\mathtt{b}, ?\mathtt{g})\}$
  : del $\{\mathtt{at}(?\mathtt{b}, ?\mathtt{r}),\ \mathtt{free}(?\mathtt{g})\}$

$\mathtt{drop}$: params $\{?\mathtt{b} : \mathtt{ball}, ?\mathtt{r} : \mathtt{room}, ?\mathtt{g} : \mathtt{gripper}\}$
  : pre $\{\mathtt{carry}(?\mathtt{b}, ?\mathtt{g}), \mathtt{at\_robby}(?\mathtt{r})\}$
  : add $\{\mathtt{at}(?\mathtt{b}, ?\mathtt{r}),\ \mathtt{free}(?\mathtt{g}))\}$
  : del $\{\mathtt{carry}(?\mathtt{b}, ?\mathtt{g})\}$

```
(: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))
)
```

groundings
= # balls × # rooms × #grippers

action labels ≈ # parameter $^{\#\ objects}$

Mutually Exclusive Actions

Know,

(at-robby room1) ⊕

(at-robby room2)

So,

(pick ball1 room1 g1) ⊕

(pick ball1 room2 g1)

```
(: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))
)
```

⊕: Mutually exclusive

Applicable Action Mutex Group (AAMG)

(pick ball1 room1 g1),
(pick ball1 room2 g1),
(pick ball1 room3 g1),
.
.
.
⊕

(pickup ball1 ␣ g1)

(pick ball2 room1 g1),
(pick ball2 room2 g1),
(pick ball2 room3 g1),
.
.
.
⊕

(pickup ball2 ␣ g1)

(pick ?b - ball ?r – room ?g – gripper)

↓

(pick ?b – ball ␣ ?g - gripper)

Seed parameters of pick:
{?b – ball , ?g – gripper}

How to identify parameter seeds?

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 $\text{seed}_x := \langle \text{seed}_x, log(|\mathcal{D}(x)|), \emptyset, \{\text{mark}(x)\}, \emptyset \rangle$
  2. get_l actions are defined for each relevant LMG l as $\text{get}_l := \langle get_l, 0, \{\text{mark}(x) \mid x \in v^f(l)\}, \{\text{mark}(y) \mid y \in v^c(l)\}, \emptyset \rangle$.
- 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

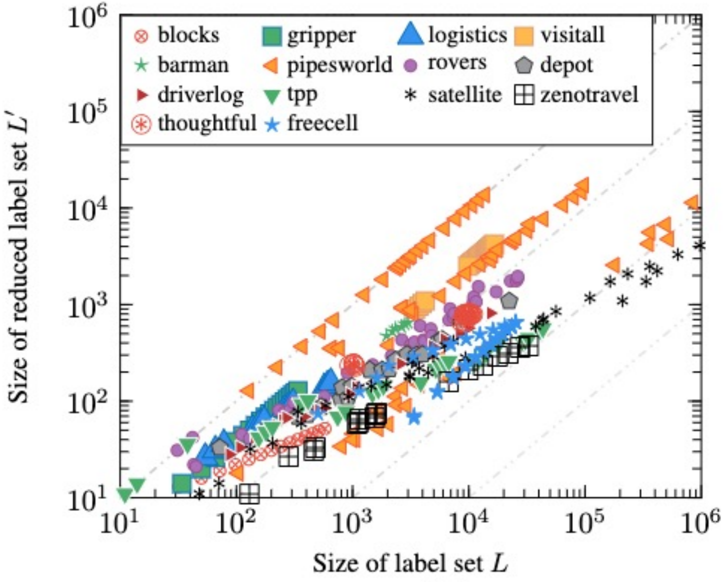
Relevant LMG Conditions

1. atom of LMG is part of precondition
2. variable types in LMG is super-type of variable of action parameter type

Contributions

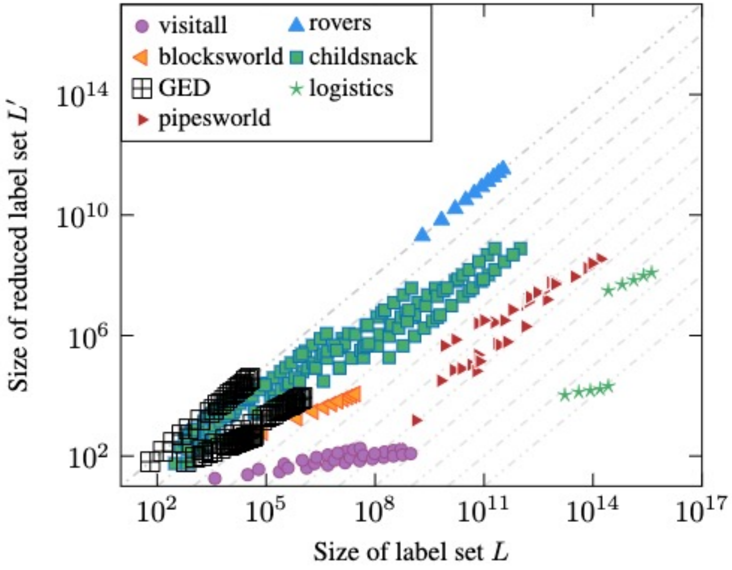
1. Characterize a *valid* label reduction
2. Propose automatic *valid* label reduction approach
   - Demonstrate that our approach achieves a significant reduction
3. Two use-cases
   - Model-free reinforcement learning
   - Lifted successor generation

Action Space Reduction



(a)

14 IPC domains



(b)

7 HTG domains

Contributions

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

Impact on learning RL policies

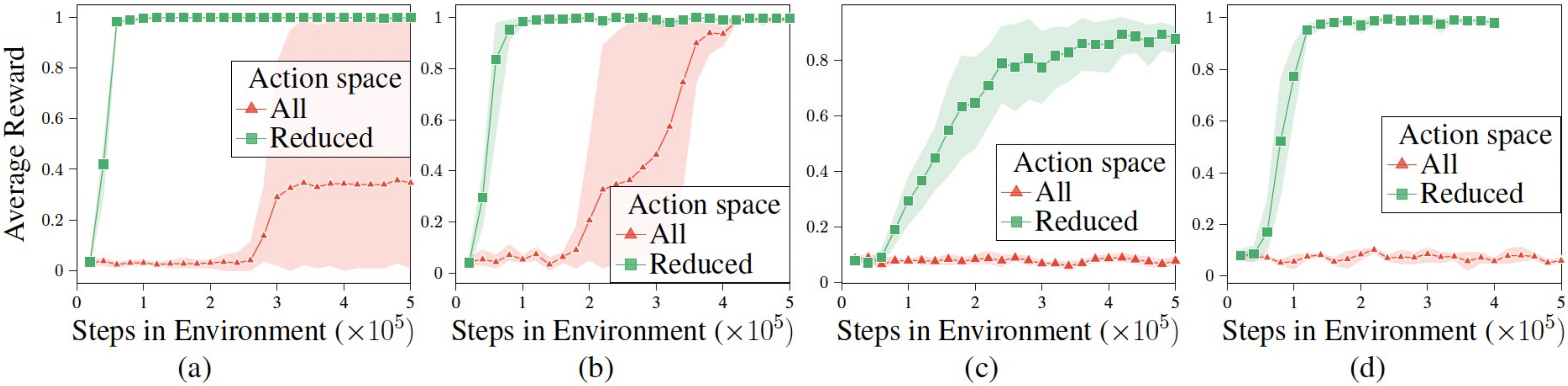


Figure 3: Learning curve in the (a) ferry, (b) gripper, (c) blocks, and (d) logistics; with and without action label reduction.

Lifted Successor Generator

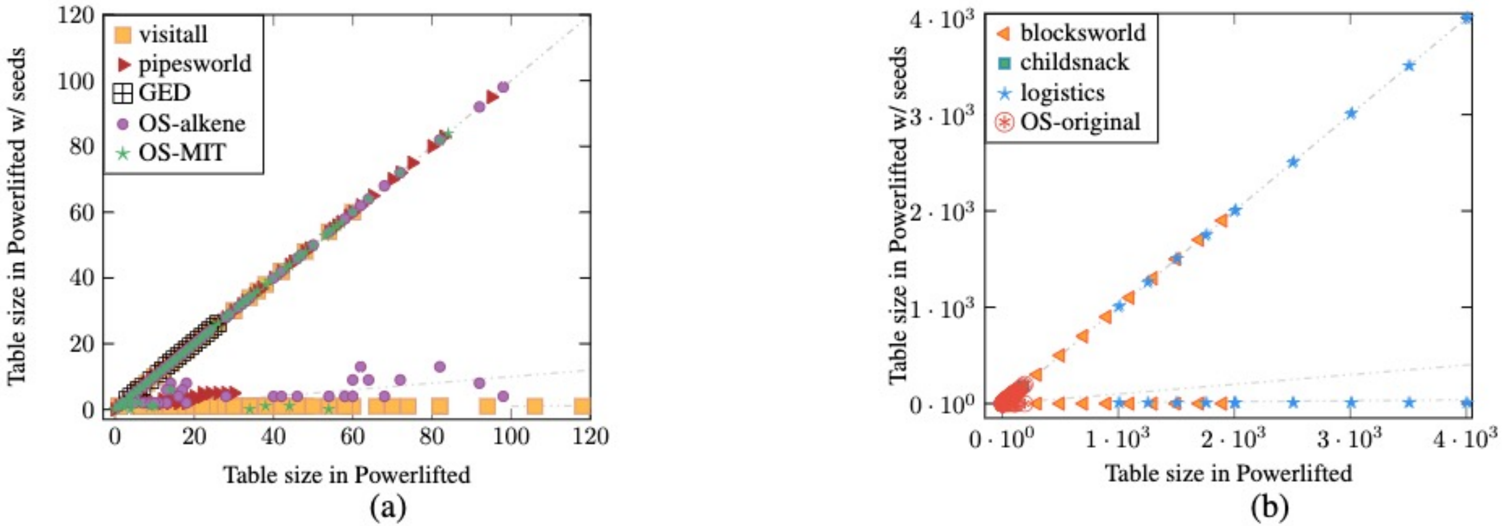


Figure 4: Comparison of table sizes before the query is performed. We split HTG domains into two plots for readability.

Contributions

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

Questions?

@ harsha_kokel

@ harsha.kokel@ibm.com

http://ibm.github.io/Parameter-Seed-Set