Operator Mutexes and Symmetries for Simplifying Planning Tasks

Daniel Fišer[1] Álvaro Torralba[2] Alexander Shleyfman[3]

[1]Czech Technical University in Prague, *danfis@danfis.cz*

[2]Saarland University, *torralba@cs.uni-saarland.de*

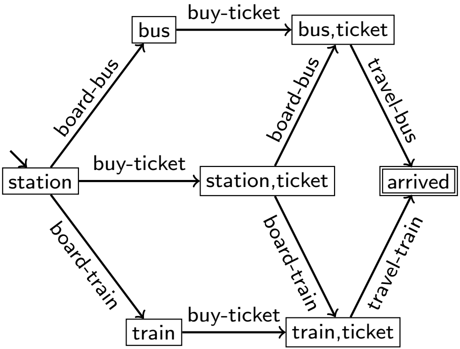
[3]Technion – Israel Institute of Technology, *alesh@campus.technion.ac.il*

January 28, 2019

STRIPS

A planning task in STRIPS is a tuple $\Pi = \langle \mathcal{F}, \mathcal{O}, I, G \rangle$ where:

- $\mathcal{F}$ is a finite set of **facts**;
- $I \subseteq \mathcal{F}$ is an **initial state**; $G \subseteq \mathcal{F}$ is a **goal specification**;
- $\mathcal{O}$ is a set of **operators** o specified by $\mathrm{pre}(o), \mathrm{del}(o), \mathrm{add}(o) \subseteq \mathcal{F}$, and $\mathrm{c}(o) \in \mathbb{R}_0^+$;



Redundancy

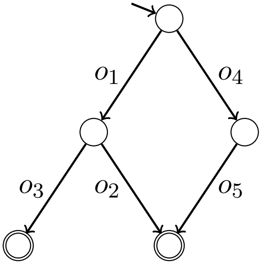
Redundant Set

A set of operators O is **redundant** if remove O from the planning task preserves at least one (strongly optimal) plan.

- Given two redundant sets O_1 and O_2, $O_1 \cup O_2$ is not necessarily redundant.
- If O_1 is redundant in Π and O_2 is redundant in $\Pi \setminus O_1$, then $O_1 \cup O_2$ is redundant in Π. Therefore, we can apply fixpoint computation.

Plans: $\langle o_1, o_2 \rangle$
$\langle o_1, o_3 \rangle$
$\langle o_4, o_5 \rangle$

Redundant sets:
$\{o_1, o_2\}$
$\{o_1, o_3\}$
$\{o_4, o_5\}$



Redundancy

Redundant Set

A set of operators O is **redundant** if remove O from the planning task preserves at least one (strongly optimal) plan.

- Given two redundant sets O_1 and O_2, $O_1 \cup O_2$ is not necessarily redundant.
- If O_1 is redundant in Π and O_2 is redundant in $\Pi \setminus O_1$, then $O_1 \cup O_2$ is redundant in Π. Therefore, we can apply fixpoint computation.

Plans: $\langle o_1, o_2 \rangle$

$\langle o_1, o_3 \rangle$

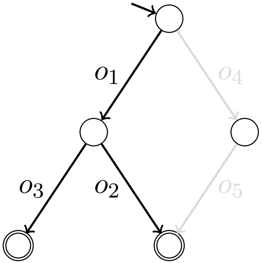
~~$\langle o_4, o_5 \rangle$~~

Redundant sets:

$\{o_2\}$

$\{o_3\}$

~~$\{o_4, o_5\}$~~



Redundancy

Redundant Set

A set of operators O is **redundant** if remove O from the planning task preserves at least one (strongly optimal) plan.

- Given two redundant sets O_1 and O_2, $O_1 \cup O_2$ is not necessarily redundant.
- If O_1 is redundant in Π and O_2 is redundant in $\Pi \setminus O_1$, then $O_1 \cup O_2$ is redundant in Π. Therefore, we can apply fixpoint computation.

Plans: $\langle o_1, o_2 \rangle$

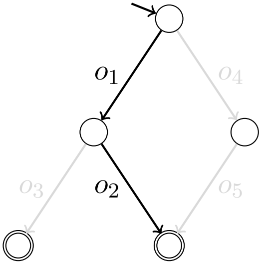
~~$\langle o_1, o_3 \rangle$~~

~~$\langle o_4, o_5 \rangle$~~

Redundant sets:

~~$\{o_3\}$~~

~~$\{o_4, o_5\}$~~



$\{o_3, o_4, o_5\}$ is redundant in Π

Operator Mutex

Operator Mutex

A **strong operator mutex** (op-mutex) O is a nonempty set of operators s.t. $O \not\subseteq \pi$ for every strongly optimal plan π.

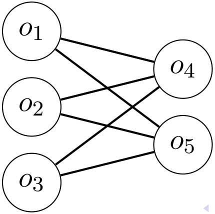
- Mutex: a set of facts M s.t. $M \not\subseteq s$ for every reachable state s.
- What is a mutex with respect to reachable states that is a strong operator mutex with respect to strongly optimal plans.
- Definition with respect to strongly optimal plans is more general than with respect to all plans.

Operator Mutex

A **strong operator mutex** (op-mutex) O is a nonempty set of operators s.t. $O \not\subseteq \pi$ for every strongly optimal plan π.

- In every op-mutex $O \subseteq \mathcal{O}$, there is an operator $o \in O$ such that $\{o\}$ is redundant.
- Given $O_1, O_2 \subseteq \mathcal{O}, O_1 \neq O_2$, if $\{o_1, o_2\}$ is an op-mutex for every $o_1 \in O_1$ and every $o_2 \in O_2$, then O_1 or O_2 is redundant.
- We proposed four different methods for inference of op-mutexes.

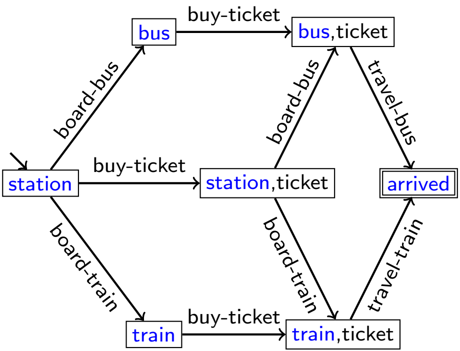
$$O_1 = \{o_1, \ldots, o_3\}$$
$$O_2 = \{o_4, \ldots, o_5\}$$



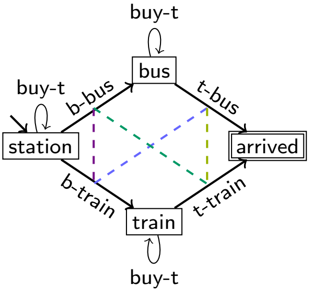
Inference of Operator Mutexes using Abstractions

Inference from Abstract State Space

Given an abstract state space Θ_Π^α and operators $o_1, o_2 \in \mathcal{O}$, $o_1 \neq o_2$, if o_2 is not reachable after o_1 and o_1 is not reachable after o_2, then $\{o_1, o_2\}$ is an op-mutex.



Projection to {station,bus,train,arrived}:



Inference of Operator Mutexes using Abstractions

Inference from Abstract State Space

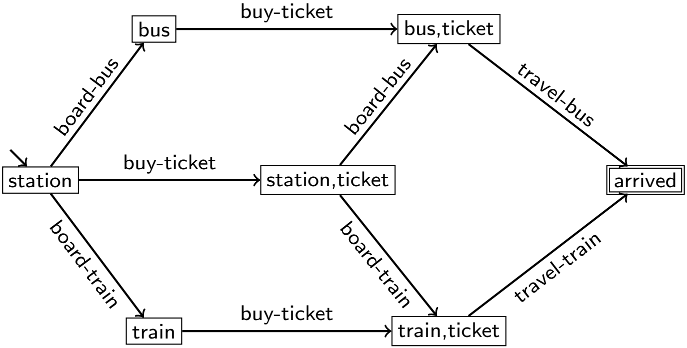
Given an abstract state space Θ_{Π}^{α} and operators $o_1, o_2 \in \mathcal{O}$, $o_1 \neq o_2$, if o_2 is not reachable after o_1 and o_1 is not reachable after o_2, then $\{o_1, o_2\}$ is an op-mutex.

- Experimentally evaluated with projections to single fact-alternating mutex groups.
- Op-mutexes were found in 32 out of 83 domains from IPC 2006–2018.
- However, the method can be used with any abstraction (Pattern Databases, Merge-and-Shrink, Cartesian Abstractions).

Operators-as-Facts Compilation

Inference with Op-Fact Compilation

Given the planning task Π, create a new auxiliary fact f_o for each operator o, and add f_o to o's add effect.
If $\{f_{o_1}, \ldots, f_{o_n}\}$ is a mutex in op-fact compilation, then $\{o_1, \ldots, o_n\}$ is an op-mutex in Π.

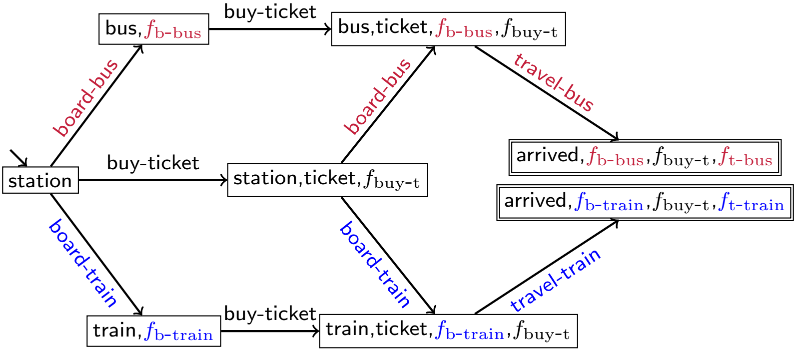


Operators-as-Facts Compilation

Inference with Op-Fact Compilation

Given the planning task Π, create a new auxiliary fact f_o for each operator o, and add f_o to o's add effect.

If $\{f_{o_1}, \ldots, f_{o_n}\}$ is a mutex in op-fact compilation, then $\{o_1, \ldots, o_n\}$ is an op-mutex in Π.



Operators-as-Facts Compilation

Inference with Op-Fact Compilation

Given the planning task Π, create a new auxiliary fact f_o for each operator o, and add f_o to o's add effect.
If $\{f_{o_1}, \dots, f_{o_n}\}$ is a mutex in op-fact compilation, then $\{o_1, \dots, o_n\}$ is an op-mutex in Π.

- Any method for inference of mutexes can be used, including h^m family of heuristics.
- Experimentally evaluated with h^2.
- Op-mutexes were found in 34 out of 83 domains from IPC 2006–2018.
- Two more methods for inference of op-mutexes can be found in the paper.

Symmetries

Definition (Symmetry)

A **plan preserving symmetry** of the transition system $\Theta = \langle \mathcal{S}, L, T, s_I, S_\star \rangle$ is a permutation σ of $\mathcal{S} \cup L$ mapping states to states and labels to labels s. t.

- $s \xrightarrow{o} s' \in T$ iff $\sigma(s) \xrightarrow{\sigma(o)} \sigma(s') \in T$,
- $\mathrm{c}(o) = \mathrm{c}(\sigma(o))$,
- $s \in S_\star$ iff $\sigma(s) \in S_\star$, and
- $\sigma(s_I) = s_I$

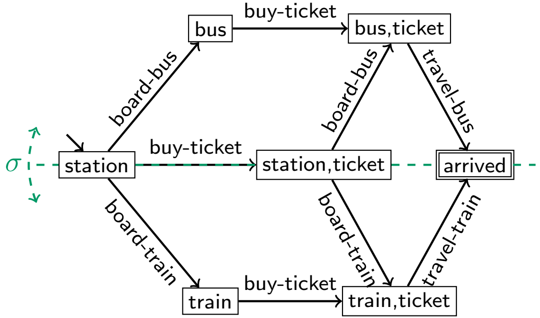
for all states $s, s' \in \mathcal{S}$ and all labels $o \in L$.

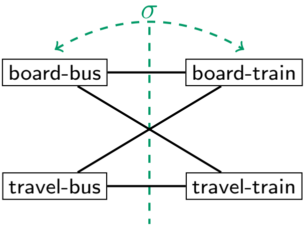
For every plan π it holds that $\sigma(\pi)$ is also a plan, moreover $|\pi| = |\sigma(\pi)|$ and $\mathrm{c}(\pi) = \mathrm{c}(\sigma(\pi))$.

Pruning with Operator Mutexes and Symmetries

Proving Redundancy

- If $\{o_1, o_2\}$ is an op-mutex and there exists a symmetry $\sigma(o_1) = o_2$, then both $\{o_1\}$ and $\{o_2\}$ are redundant.
- And similarly for two sets of operators O_1 and O_2: If $\{o_1, o_2\}$ is an op-mutex for every $o_1 \in O_1$ and every $o_2 \in O_2$, and there exists symmetry $\sigma(O_1) = O_2$, then both O_1 and O_2 are redundant.

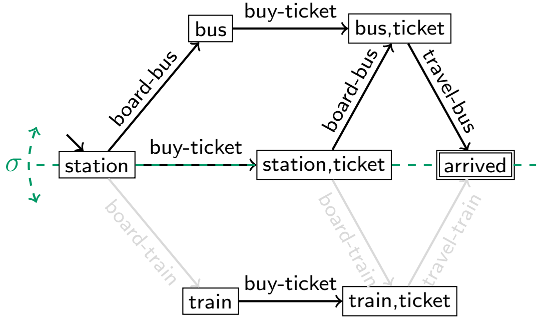


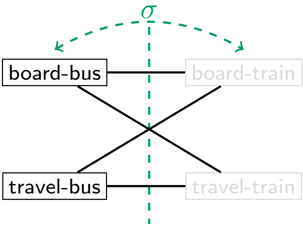


Pruning with Operator Mutexes and Symmetries

Proving Redundancy

- If $\{o_1, o_2\}$ is an op-mutex and there exists a symmetry $\sigma(o_1) = o_2$, then both $\{o_1\}$ and $\{o_2\}$ are redundant.
- And similarly for two sets of operators O_1 and O_2: If $\{o_1, o_2\}$ is an op-mutex for every $o_1 \in O_1$ and every $o_2 \in O_2$, and there exists symmetry $\sigma(O_1) = O_2$, then both O_1 and O_2 are redundant.





Pruning with Operator Mutexes and Symmetries

Proving Redundancy

- If $\{o_1, o_2\}$ is an op-mutex and there exists a symmetry $\sigma(o_1) = o_2$, then both $\{o_1\}$ and $\{o_2\}$ are redundant.
- And similarly for two sets of operators O_1 and O_2: If $\{o_1, o_2\}$ is an op-mutex for every $o_1 \in O_1$ and every $o_2 \in O_2$, and there exists symmetry $\sigma(O_1) = O_2$, then both O_1 and O_2 are redundant.

- Each pruning step destroys at least one symmetry, therefore the number of steps is limited by the number of symmetries.
- Finding a complete bipartite subgraph is NP-hard.
- Therefore, we use a greedy approach for inference of redundant sets: We choose the largest set of operators that destroys the least number of symmetries in a hope that we will be able to do more steps.

Pruning Results

domain	baseline **h^2+de**	**fam**	Π^2_{op}	**h^2+de with** **fam**	**h^2+de with** Π^2_{op}
barman11	**52.11**	4.71	24.48	+**3.60**	+**3.60**
barman14	**53.43**	4.94	25.28	+**4.01**	+**4.01**
caldera18	**39.34**	0.73	13.25	0.00	+**1.45**
cavediving14	0.66	**1.15**	**1.44**	+**1.15**	+**1.15**
childsnack14	0.00	**39.58**	**39.58**	+**39.58**	+**39.58**
hiking14	0.00	**0.07**	**0.07**	+**0.07**	+**0.07**
parcprinter08	**70.68**	28.61	28.61	0.00	0.00
parcprinter11	**70.15**	25.75	25.75	0.00	0.00
pathways06	**3.94**	2.22	2.26	+**3.73**	+**3.73**
pegsol08	**13.57**	0.00	1.92	+**0.36**	+**0.36**
pegsol11	**9.53**	1.08	3.51	+**0.20**	+**0.20**
pipesworld06	**11.46**	0.04	4.04	+**0.04**	+**0.04**
scanalyzer08	**3.19**	0.00	2.10	0.00	0.00
scanalyzer11	**3.17**	0.00	2.12	0.00	0.00
sokoban08	0.24	1.54	**1.71**	+1.54	+**1.60**
sokoban11	0.35	1.86	**2.12**	+1.86	+**1.95**
tpp06	**37.96**	1.49	19.97	+**1.79**	+**1.79**
trucks06	**75.02**	0.00	0.71	+**7.87**	+**7.87**
woodworking08	**53.47**	2.83	10.79	+**1.60**	+**1.60**
woodworking11	**53.72**	2.73	10.17	+**1.83**	+**1.83**
overall	**53.72**	7.99	13.73	+4.29	+**4.39**

Table: Average percentage of removed operators with 15 minutes time limit.

Conclusion

- We intrdouced a new concept, called operator mutex, stating that a certain set of operators cannot appear together in the same plan.
- We proposed four different methods for inference of op-mutexes.
- We showed that even the simplest variants of the inference methods were able to find op-mutexes in a sizeable amount of domains.
- We combined op-mutexes with structural symmetries to prove certain sets of operators to be redundant so they can be safely removed from the planning task.