Factored Symmetries for Merge-and-Shrink Abstractions

Silvan Sievers[1], Martin Wehrle[1], Malte Helmert[1], Alexander Shleyfman[2] and Michael Katz[3]
[1]University of Basel, Switzerland [2]Technion, Haifa, Israel [3]IBM Haifa Research Lab, Israel

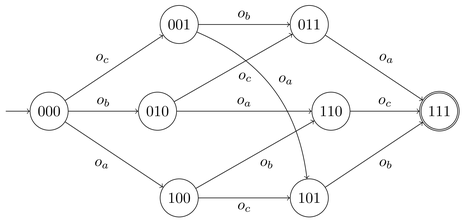
Introduction

Two existing successful techniques:
- Merge-and-shrink heuristics: state-of-the-art abstraction heuristic for classical planning
- Symmetry elimination: prominent example of a state space pruning technique

Contribution: use symmetries to construct better merge-and-shrink heuristics

Classical Planning

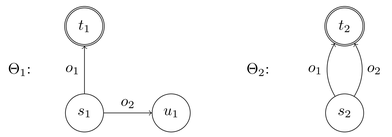
Planning task:
- State variables
- Initial state
- Goal description
- Operators



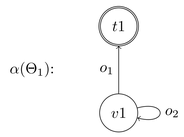
Merge-and-Shrink

1 $\mathcal{T} := \{$atomic transition systems$\}$
2 While $|\mathcal{T}| > 1$:
3 Choose $\Theta_1, \Theta_2 \in \mathcal{T}$ according to merging strategy
4 Shrink according to shrinking strategy: $\Theta_1 := \alpha_1(\Theta_1)$, $\Theta_2 := \alpha_2(\Theta_2)$
5 Merge by computing the synchronized product: $\Theta_1 \otimes \Theta_2$
6 Replace Θ_1 and Θ_2 by $\Theta_1 \otimes \Theta_2$ in $\mathcal{T}$

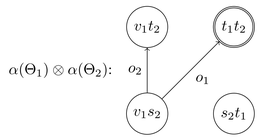
Atomic transition systems:



Shrink Θ_1 with $\alpha(s_1) = \alpha(u_1)$ (and Θ_2 with $\alpha = \text{id}$):



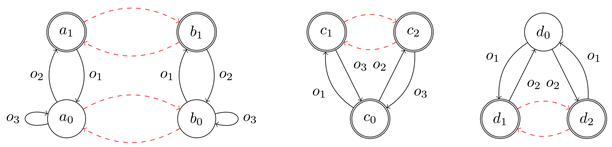
Merge $\alpha(\Theta_1)$ and $\alpha(\Theta_2)$:



Factored Symmetries

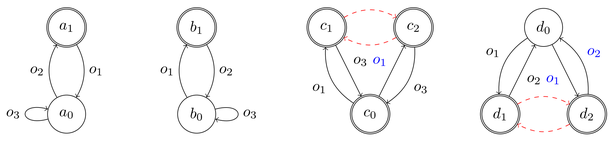
General:

$\sigma(o_1) = o_2$
$\sigma(o_2) = o_1$
$\sigma(o_3) = o_3$



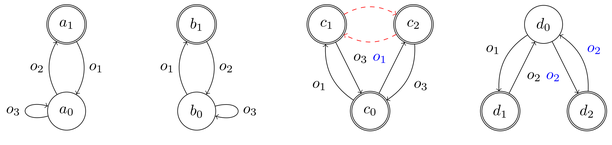
Local:

$\sigma(o_1) = o_1$
$\sigma(o_2) = o_2$
$\sigma(o_3) = o_3$



Atomic:

$\sigma(o_1) = o_1$
$\sigma(o_2) = o_2$
$\sigma(o_3) = o_3$



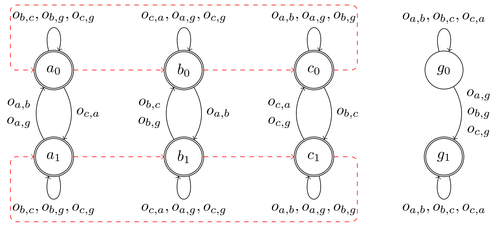
Interaction of Symmetries with Merge-and-Shrink

Interaction of symmetries with shrinking:
- Can break existing symmetries (e.g. combining a_0 and a_1 in example below)
- Can create new symmetries by removing obstacles to symmetries between states

Interaction of symmetries with merging:
- Can break existing symmetries (e.g. merging the two left-most transition systems in example below breaks symmetry between $a_1b_0c_0g_0$ and $a_0b_0c_1g_0$)
- Can create new symmetries (e.g. recover the same symmetry by merging the product with the third transition system)

Operators:

$\sigma(o_{a,b}) = o_{b,c}$
$\sigma(o_{b,c}) = o_{c,a}$
$\sigma(o_{c,a}) = o_{a,b}$



Factored symmetry example: rotating $a \mapsto b \mapsto c \mapsto a$

Factored Symmetries and Merge-and-Shrink: Shrinking

Can we use factored symmetries for shrinking?

Proposition

Shrinking based on local symmetries is not information-preserving.

Proposition

Shrinking based on atomic symmetries is captured by shrinking based on bisimulation in combination with full (exact) label reduction.

Corollary

Shrinking based on atomic symmetries is information-preserving.

Factored Symmetries and Merge-and-Shrink: Merging

Can we use factored symmetries for merging?

Proposition

Merging all transition systems affected by a local non-atomic symmetry gives rise to an atomic symmetry.

Symmetry-Enhanced Merge-and-Shrink

Our integration of symmetries into merge-and-shrink:
- Atomic symmetries implicitly captured by bisimulation
- Compute non-atomic symmetries and merge affected transition systems

Algorithm:
1 $\mathcal{T} := \{$atomic transition systems$\}$
2 $N = \emptyset$
3 While $|\mathcal{T}| > 1$:
4 If $|N| \leq 1$:
5 Compute a set Σ of non-atomic symmetries of $\mathcal{T}$.
6 If $\Sigma \neq \emptyset$:
7 Let $N := \{\Theta \in \mathcal{T} \mid \Theta$ is affected by one chosen $\sigma \in \Sigma\}$
8 If $|N| \geq 2$:
9 Choose $\Theta_1, \Theta_2 \in N$.
10 else
11 Choose Θ_1, Θ_2 according to basic merging strategy M.
12 Apply shrinking w.r.t. basic shrinking strategy S on Θ_1, Θ_2.
13 Replace Θ_1, Θ_2 by $\Theta_1 \otimes \Theta_2$ in $\mathcal{T}$ and in N (if applicable) .

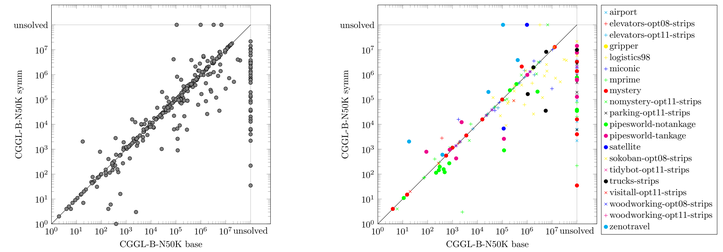
General Results for Various Merging Strategies

	Coverage (blind search: 519)			**#successful M&S constr.**		
	base	symm	symm-1	base	symm	symm-1
CGGL-B-N50K	600	646	637	1138	1177	1181
DFP-B-N50K	644	657	646	1181	1204	1203
MIASM-B-N50K	654	659	660	1159	1162	1172
RL-B-N50K	634	652	643	1202	1219	1216
RND-B-N50K	583	622	605	1165	1207	1200

- All configurations: shrinking based on bisimulation, size limit of 50000 states
- base: baseline without symmetries
- symm: compute symmetries up to a total time limit of 60 seconds
- symm-1: compute symmetries only once on atomic transition systems

Detailed Results for CGGL Merging

Coverage	CGGL base	CGGL symm
gripper (20)	7	+11
parking-opt11-strips (20)	0	+7
mystery (30)	12	+5
pipesworld-tankage (50)	9	+5
airport (50)	11	+4
miconic (150)	74	+4
mprime (35)	20	+3
pipesworld-notankage (50)	12	+3
sokoban-opt08-strips (30)	27	+3
elevators-opt08-strips (30)	12	+1
elevators-opt11-strips (20)	10	+1
trucks-strips (30)	7	+1
visitall-opt11-strips (20)	9	+1
woodworking-opt08-strips (30)	11	+1
woodworking-opt11-strips (20)	6	+1
logistics98 (35)	5	-1
nomystery-opt11-strips (20)	19	-1
satellite (36)	7	-1
tidybot-opt11-strips (20)	1	-1
zenotravel (20)	11	-1
Sum (716)	270	+46
Remaining domains (680)	330	±0
Sum (1396)	600	646



Expansions for CGGL base vs. symm (right: restricted to domains with different coverage)

Outcome	base	symm	symm-1
M&S out of memory	143	100	96
M&S out of time	115	119	119
Search out of memory	530	524	536
Search out of time	4	2	3
Proved unsolvable	4	5	5
Solved	600	646	637

Planner outcome (reasons for termination)

Outcome	vs. symm	vs. symm-1
M&S out of memory	27	23
M&S out of time	0	0
Search out of memory	25	19
Search out of time	0	0
Solved	594	595

Outcome of CGGL base on tasks solved by CGGL symm/CGGL symm-1

Contributions

- Introduced notion of factored symmetries
- Merging and shrinking can lead to the loss and the discovery of symmetries
- Symmetry-enhanced merging strategies increase performance of merge-and-shrink heuristics