Generalized Label Reduction for Merge-and-Shrink Heuristics

Silvan Sievers and Martin Wehrle and Malte Helmert
University of Basel, Switzerland

Motivation

State-space search:
- Fundamental problem of artificial intelligence
- Specify large state spaces compactly as family of labeled transition systems

Merge-and-shrink heuristic:
- Construct single transition system starting from a family of small transition systems
- State-of-the-art abstraction heuristic in planning

Labeled Transition Systems

A labeled transition system is a 4-tuple $\Theta = \langle S, L, T, S_\star \rangle$ with
- S: a finite set of states
- L: a finite set of labels
- $T \subseteq S \times L \times S$: a set of (labeled) transitions
- $S_\star \subseteq S$: a set of goal states

Notation:
- $s \xrightarrow{\ell} s'$ for transition $\langle s, \ell, s' \rangle \in T$
- $s \xrightarrow{\ell} s' \in \Theta$ for $s \xrightarrow{\ell} s' \in T$ with T transition relation of Θ

Merge-and-Shrink Heuristics

Computation of merge-and-shrink heuristics:
- Start with the set X of atomic transition systems
- Transform X by repeatedly applying one of the following:
  - Merge: replace two transition systems $\Theta, \Theta' \in X$ by their synchronized product
  - Shrink: replace a transition system $\Theta \in X$ by an abstract transition system
- Stop when one transition system is left, use as heuristic

Label Reduction for Merge-and-Shrink

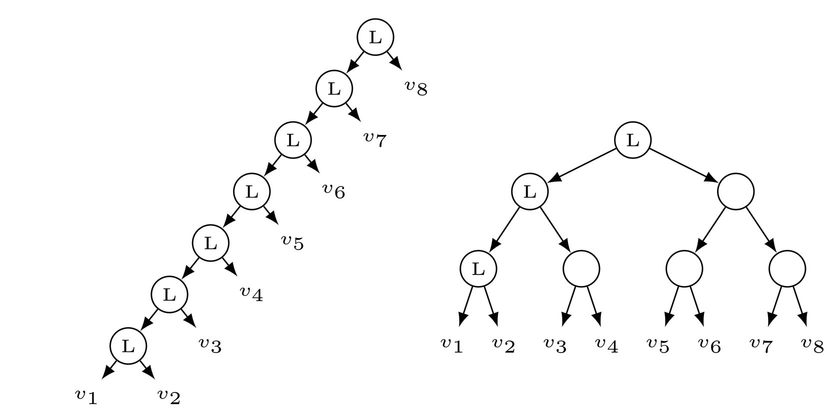
Concept:
- Identify and eliminate semantically equivalent labels in transition systems
- Always useful:
  - Reduction of memory and time consumption
  - Heuristic quality preserved
  - Fast to compute
- Crucial for efficiently computing merge-and-shrink heuristics

Previous Label Reduction in the Merge-and-Shrink Computation

Previous theory:
- Choose one pivot variable
- Label reduction only allowed for transition systems containing pivot variable

Consequence:
- Label reduction only possible in one branch of the merge tree

Example merge trees:



Discussion of Previous Label Reduction

Drawbacks:
- Local transformation of one transition system (problematic for synchronization behavior)
- Syntax-based comparison of labels (requires access to underlying planning operators)
- Independence of shrink strategy (no label reduction opportunities from shrinking)

Consequences:
- Label reduction only applicable in limited cases (pivot variable)
- Rather complex theory
- Usage of linear merge strategies to circumvent drawbacks
- Large part of the space of possible merge strategies not yet explored

Generalized Label Reduction

Definition

Let X be a set of transition systems with label set L.
A label reduction for X is defined as follows:
- For a set of labels $L' \subseteq L$, choose new label $\ell \notin L$ with cost $cost(\ell) := min_{\ell' \in L'} cost(\ell')$.
- Replace each label $\ell' \in L'$ by the new label ℓ in all transition systems.

Formally: a label reduction τ is a label mapping, i. e. a function defined on L.

Theorem

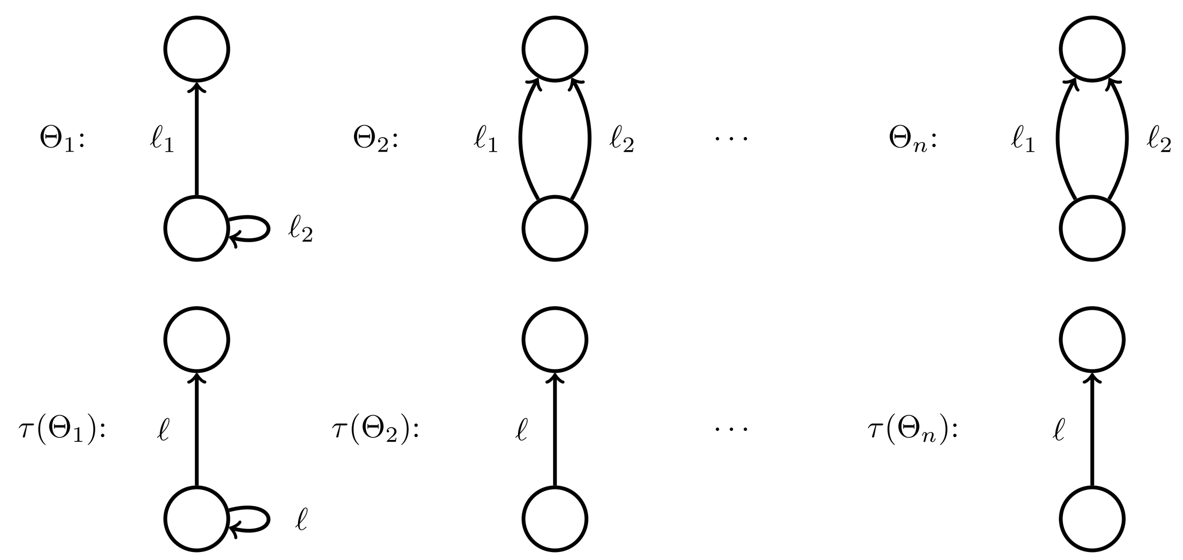
Label reduction is always safe, i. e. leaves the heuristic admissible.

Intuition:
- Synchronization behavior preserved because transitions are preserved
- (Goal) states of transition systems not modified
- Transition costs not increased

Example:

Before label reduction:

After label reduction:



Main Result

Let $X = \{\Theta_1, \dots \Theta_n\}$ be a set of transition systems with label set L and let $\ell_1, \ell_2 \in L$.

Terminology

Definitions:
- ℓ_1 and ℓ_2 are locally equivalent in Θ_i if they label the same set of transitions in Θ_i.
- ℓ_1 and ℓ_2 are Θ_i-combinable in X if they are locally equivalent in all $\Theta_j \in X \setminus \{\Theta_i\}$.
- ℓ_1 globally subsumes ℓ_2 if the set of transitions labeled by ℓ_2 is a subset of the set of transitions labeled by ℓ_1 in all transition systems.

Example: ℓ_1 and ℓ_2 in example above are Θ_1-combinable.

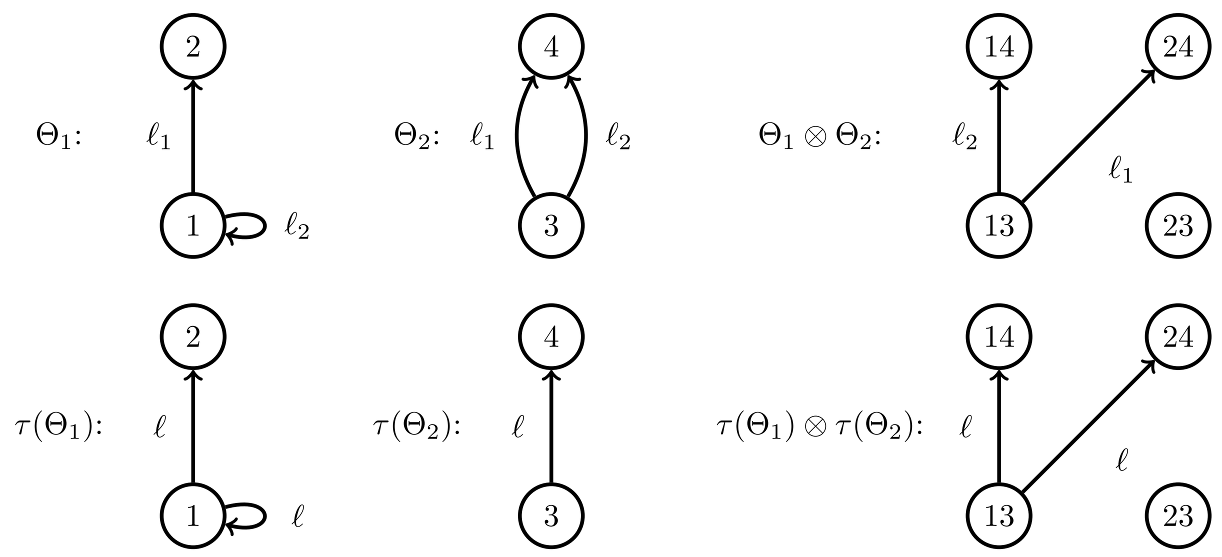
Theorem

Let τ be a label reduction which combines labels ℓ_1 and ℓ_2 and replaces them by some label ℓ and leaves all other labels unchanged. τ is exact, i. e. leaves the heuristic perfect, iff $cost(\ell_1) = cost(\ell_2)$ and
1. ℓ_1 globally subsumes ℓ_2, or
2. ℓ_2 globally subsumes ℓ_1, or
3. ℓ_1 and ℓ_2 are Θ-combinable for some $\Theta \in X$.

Example:

Merge without label reduction:

Merge with label reduction:



Experimental Setup

Evaluated merge-and-shrink strategies:
- Linear merge strategy reverse-level (RL)
- Non-linear merge strategy proposed by Dräger et al. (DFP)
- Shrinking based on bisimulation (B) and greedy bisimulation (G)

Coverage Results

Observations:
- Label reduction always useful
- New label reduction better than old one
- Non-linear merge strategy DFP best performer

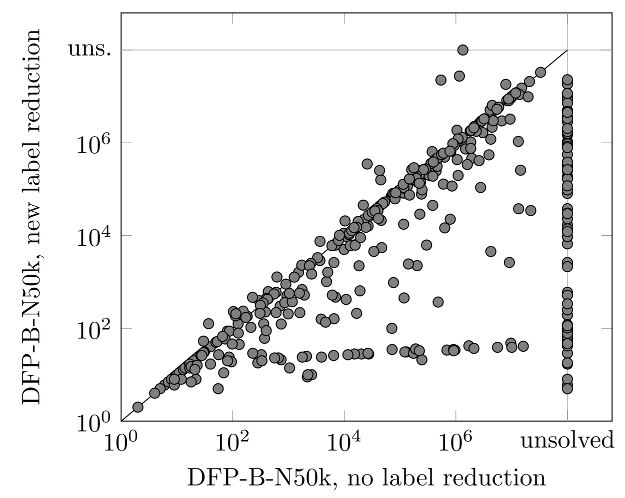
Coverage:

merge/shrink strategy	none	old	new
RL-G-N∞	417	485	465
RL-B-N10k	590	624	617
RL-B-N50k	577	618	634
RL-B-N100k	560	599	639
RL-B-N200k	544	590	630
RL-B-N∞	257	302	302
DFP-G-N∞	415	—	465
DFP-B-N10k	597	—	622
DFP-B-N50k	565	—	644
DFP-B-N100k	551	—	632
DFP-B-N200k	522	—	625
DFP-B-N∞	253	—	302

Results: Usefulness of Label Reduction

Remarks:
- Label reduction of crucial importance for efficiency
- Better informed heuristics because bisimulation shrinking profits from label reduction

Expansions:



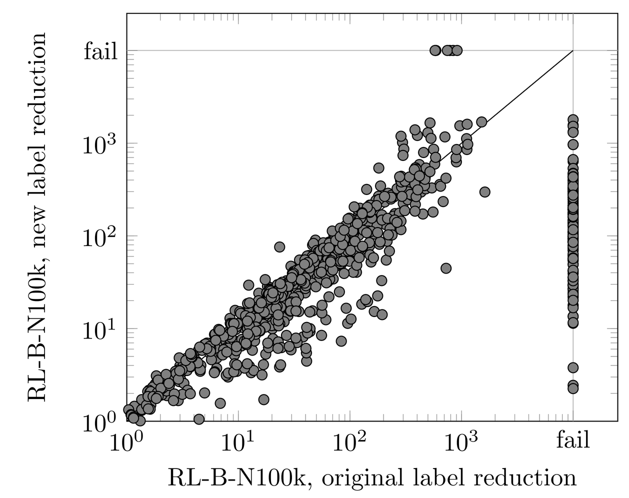
Coverage:

	RL-B-100K			DFP-B-50K	
	none	old	new	none	new
mprime (35)	8	+6	+15	6	+17
miconic (150)	60	+13	+13	58	+14
gripper (20)	7	+13	+13	7	+11
freecell (80)	6	−2	+13	9	+11
mystery (30)	8	+1	+8	8	+8
zenotravel (20)	9	+3	+3	10	+2
pipesworld-tankage (50)	8	+2	+3	12	+2
nomystery-opt11-strips (20)	17	+1	+1	16	+2
woodworking-opt08-strips (30)	11	−1	+1	11	+2
blocks (35)	25	−3	−3	25	+2
grid (5)	1	+2	+2	1	+1
floortile-opt11-strips (20)	5	+1	+1	4	+1
rovers (40)	7	+1	+1	7	+1
satellite (36)	5	+1	+1	5	+1
scanalyzer-08-strips (30)	12	+1	+1	12	+1
scanalyzer-opt11-strips (20)	9	+1	+1	9	+1
woodworking-opt11-strips (20)	6	−1	+1	6	+1
pipesworld-notankage (50)	14	±0	±0	14	+1
sokoban-opt08-strips (30)	24	±0	+2	25	±0
trucks-strips (30)	6	±0	+2	6	±0
transport-opt11-strips (20)	6	+1	+1	6	±0
driverlog (20)	13	−1	−1	12	±0
Sum (**791**)	267	+39	+79	269	+79
Remaining domains (**605**)	293	±0	±0	296	±0
Sum (**1396**)	560	599	639	565	644

Results: Old vs. New Label Reduction Method

Remarks:
- Larger computational effort compensated by reduced memory/time consumption of merge-and-shrink
- Speed gain due to reduced computation effort of merge and shrink operations
- Failures almost always due to memory limit

Construction time:



Conclusion

Contributions:
- Generalized label reduction for merge-and-shrink heuristics:
  - Purely semantic operation
  - Always allowed on all transition systems
  - Safe and exact (under conditions) transformation of transition systems
- Prepared the ground for non-linear merge strategies in practice:
  - Implemented non-linear merge strategy DFP