Planning and Optimization

D10. M&S: Merging Strategies and Label Reduction

Malte Helmert and Gabriele Röger

Universität Basel

November 17, 2016

Planning and Optimization

November 17, 2016 — D10. M&S: Merging Strategies and Label Reduction

D10.1 Merging Strategies

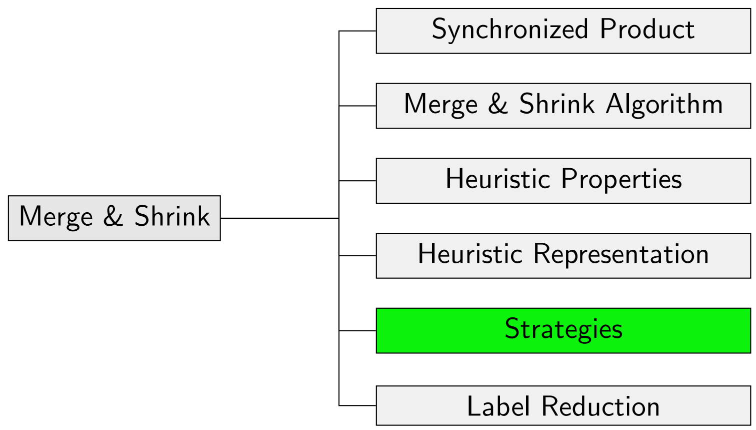
D10.2 Label Reduction

D10.3 Summary

D10.4 Literature

D10.1 Merging Strategies

Merging Strategies



Generic Algorithm Template

Generic M&S computation algorithm

abs := $\{\mathcal{T}^{\pi_{\{v\}}} \mid v \in V\}$
while *abs* contains more than one abstract transition system:
    select $\mathcal{A}_1$, $\mathcal{A}_2$ from *abs*
    shrink $\mathcal{A}_1$ and/or $\mathcal{A}_2$ until $size(\mathcal{A}_1) \cdot size(\mathcal{A}_2) \leq N$
    $abs := abs \setminus \{\mathcal{A}_1, \mathcal{A}_2\} \cup \{\mathcal{A}_1 \otimes \mathcal{A}_2\}$
return the remaining abstract transition system in *abs*

Remaining question:

- Which abstractions to select? $\rightsquigarrow$ merging strategy

Linear Merging Strategies

Linear Merging Strategy

In each iteration after the first, choose the abstraction computed in the previous iteration as $\mathcal{A}_1$.

Rationale: only maintains one "complex" abstraction at a time

$\rightsquigarrow$ Fully defined by an ordering of atomic projections.

Linear Merging Strategies: Choosing the Ordering

Use similar causal graph criteria as for growing patterns.

Example: Strategy of h_{HHH}

h_{HHH}: Ordering of atomic projections

- Start with a goal variable.
- Add variables that appear in preconditions of operators affecting previous variables.
- If that is not possible, add a goal variable.

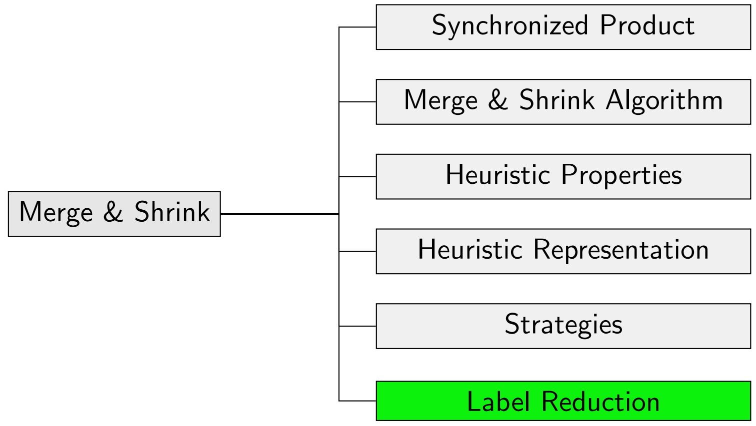
Rationale: increases h quickly

Non-linear Merging Strategies

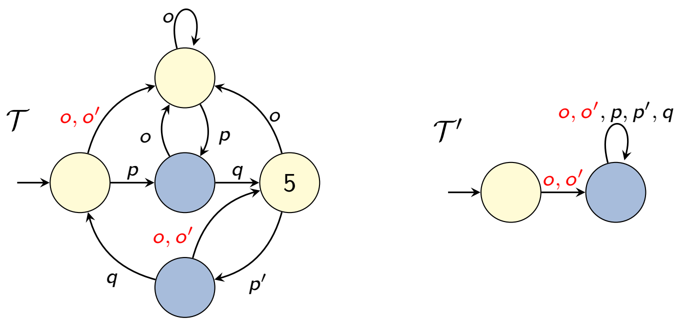
- Non-linear merging strategies only recently gained more interest in the planning community.
- One reason: Better label reduction techniques (later in this chapter) enabled a more efficient computation.
- Examples:
  - DFP: preferrably merge transition systems that must synchronize on labels that occur close to a goal state.
  - UMC and MIASM: Build clusters of variables with strong interactions and first merge variables within each cluster.
- Each merge-and-shrink heuristic computed with a non-linear merging strategy can also be computed with a linear merging strategy.
- However, linear merging can require a super-polynomial blow-up of the final representation size.

D10.2 Label Reduction

Merging Strategies



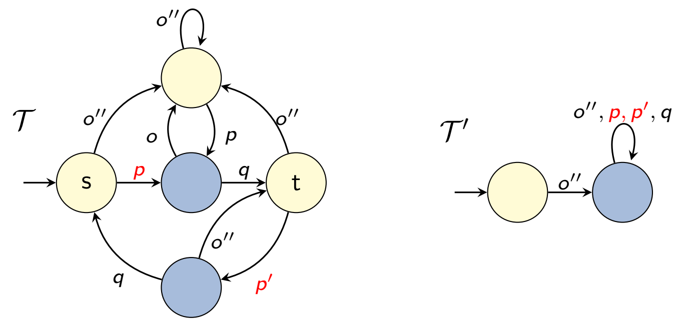
Label Reduction: Motivation (1)



Whenever there is a transition with label o' there is also a transition with label o. If o' is not cheaper than o, we can always use the transition with o.

Idea: Replace o and o' with label o'' with cost of o

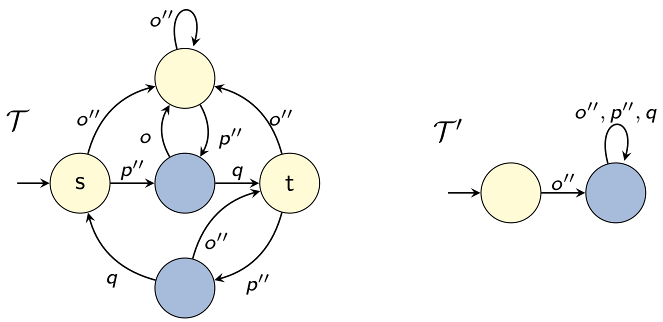
Label Reduction: Motivation (2)



States s and t are not bisimilar due to labels p and p'. In $\mathcal{T}'$ they label the same (parallel) transitions. If p and p' have the same cost, in such a situation there is no need for distinguishing them.

Idea: Replace p and p' with label p'' with same cost.

Label Reduction: Motivation (3)



Label reductions reduce the time and memory requirement for merge and shrink steps and enable coarser bisimulation abstractions.

When is label reduction a safe transformation?

Label Reduction: Definition

Definition (Label Reduction)

Let X be a collection of transition systems with label set L and label cost function c. A label reduction $\langle \lambda, c' \rangle$ for X is given by a function $\lambda : L \to L'$, where L' is an arbitrary set of labels, and a label cost function c' on L' such that for all $\ell \in L$, $c'(\lambda(\ell)) \leq c(\ell)$.

For $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle \in X$ the label-reduced transition system is $\mathcal{T}^{\langle \lambda, c' \rangle} = \langle S, L', c', \{\langle s, \lambda(\ell), t \rangle \mid \langle s, \ell, t \rangle \in T\}, s_0, S_\star \rangle$.

The label-reduced collection is $X^{\langle \lambda, c' \rangle} = \{\mathcal{T}^{\langle \lambda, c' \rangle} \mid \mathcal{T} \in X\}$.

$L' \cap L \neq \emptyset$ and $L' = L$ are allowed.

Label Reduction is Safe (1)

Theorem (Label Reduction is Safe)

Let X be a collection of transition systems and $\langle \lambda, c' \rangle$ be a label-reduction for X. The transformation from X to $X^{\langle \lambda, c' \rangle}$ is safe.

Proof.

We show that the transformation is safe, using $\sigma = \mathrm{id}$ for the mapping of states and λ for the mapping of labels.

The label cost function of $\mathcal{T}_{X^{\langle \lambda, c' \rangle}}$ is c' and has the required property by the definition of label reduction. ...

Label Reduction is Safe (2)

Theorem (Label Reduction is Safe)

Let X be a collection of transition systems and $\langle \lambda, c' \rangle$ be a label-reduction for X. The transformation from X to $X^{\langle \lambda, c' \rangle}$ is safe.

Proof (continued).

By the definition of synchronized products, $\mathcal{T}_X$ has a transition $\langle \langle s_1, \dots, s_{|X|} \rangle, \ell, \langle t_1, \dots, t_{|X|} \rangle \rangle$ if for all i, $\mathcal{T}_i \in X$ has a transition $\langle s_i, \ell, t_i \rangle$. By the definition of label-reduced transition systems, this implies that $\mathcal{T}^{\langle \lambda, c' \rangle}$ has a corresponding transition $\langle s_i, \lambda(\ell), t_i \rangle$, so $\mathcal{T}_{X^{\langle \lambda, c' \rangle}}$ has a transition $\langle s, \lambda(\ell), t \rangle = \langle \sigma(s), \lambda(\ell), \sigma(t) \rangle$ (definition of synchronized products).

For each goal state $s_\star$ of $\mathcal{T}_X$, state $\sigma(s_\star) = s_\star$ is a goal state of $\mathcal{T}_{X^{\langle \lambda, c' \rangle}}$ because the transformation replaces each transition system with a system that has the same goal states. □

More Terminology

Let X be a collection of transition systems with labels L. Let $\ell, \ell' \in L$ be labels and let $\mathcal{T} \in X$.

- Label ℓ is alive in X if all $\mathcal{T}' \in X$ have some transition labelled with ℓ. Otherwise, ℓ is dead.
- Label ℓ locally subsumes label ℓ' in $\mathcal{T}$ if for all transitions $\langle s, \ell', t\rangle$ of $\mathcal{T}$ there is also a transition $\langle s, \ell, t\rangle$ in $\mathcal{T}$.
- ℓ globally subsumes ℓ' if it locally subsumes ℓ' in all $\mathcal{T}' \in X$.
- ℓ and ℓ' are locally equivalent in $\mathcal{T}$ if they label the same transitions in $\mathcal{T}$, i.e. ℓ locally subsumes ℓ' in $\mathcal{T}$ and vice versa.
- ℓ and ℓ' are $\mathcal{T}$-combinable if they are locally equivalent in all transition systems $\mathcal{T}' \in X \setminus \{\mathcal{T}\}$.

Exact Label Reduction

Theorem (Criteria for Exact Label Reduction)

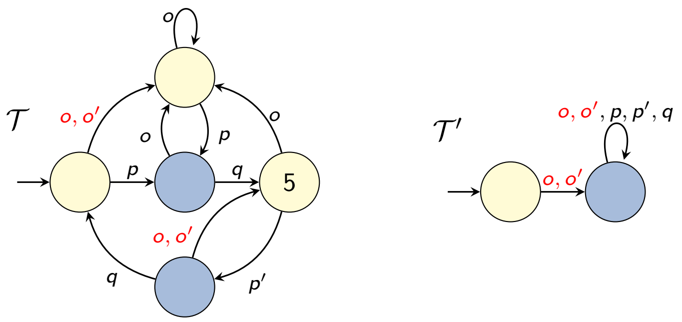
Let X be a collection of transition systems with cost function c and label set L that contains no dead labels.

Let $\langle \lambda, c'\rangle$ be a label-reduction for X such that λ combines labels ℓ_1 and ℓ_2 and leaves other labels unchanged. The transformation from X to $X^{\langle \lambda, c'\rangle}$ is exact iff $c(\ell_1) = c(\ell_2)$, $c'(\lambda(\ell)) = c(\ell)$ for all $\ell \in L$, and

- *ℓ_1 globally subsumes ℓ_2, or*
- *ℓ_2 globally subsumes ℓ_1, or*
- *ℓ_1 and ℓ_2 are $\mathcal{T}$-combinable for some $\mathcal{T} \in X$.*

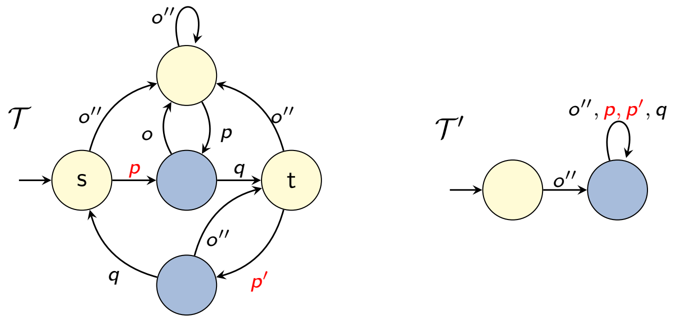
(Proof omitted.)

Back to Example (1)



Label o' globally subsumes label o.

Back to Example (2)



Labels p and p' are $\mathcal{T}$-combinable.

Computation of Exact Label Reduction (1)

- For given labels ℓ_1, ℓ_2, the criteria can be tested in low-order polynomial time.
- Finding globally subsumed labels involves finding subset relationsships in a set family.
  $\rightsquigarrow$ no linear-time algorithms known
- The following algorithm exploits only $\mathcal{T}$-combinability.

Computation of Exact Label Reduction (2)

eq_i := set of label equivalence classes of $\mathcal{T}_i \in X$

Label-reduction based on $\mathcal{T}_i$-combinability

$eq := \{L\}$
for $j \in \{1, \dots, |X|\} \setminus \{i\}$
  Refine eq with eq_j
// two labels are in the same set of eq
// iff they are locally equivalent in all $\mathcal{T}_j \neq \mathcal{T}_i$.
$\lambda = \mathrm{id}$
for $B \in eq$
  $samecost := \{[\ell]_{\sim_c} \mid \ell \in B, \ell' \sim_c \ell'' \text{ iff } c(\ell') = c(\ell'')\}$
  for $L' \in samecost$
    ℓ_{new} := new label
    $c'(\ell_{\text{new}})$:= cost of labels in L'
    for $\ell \in L'$
      $\lambda(\ell) = \ell_{\text{new}}$

Application in Merge-and-Shrink Algorithm

Generic M&S Computation Algorithm with Label Reduction

$abs := \{\mathcal{T}^{\pi_{\{v\}}} \mid v \in V\}$
while abs contains more than one abstract transition system:
  select $\mathcal{T}_1$, $\mathcal{T}_2$ from abs
  possibly label-reduce all $\mathcal{T} \in abs$
    (e.g. based on $\mathcal{T}_1$- and/or $\mathcal{T}_2$-combinability).
  shrink $\mathcal{T}_1$ and/or $\mathcal{T}_2$ until $size(\mathcal{T}_1) \cdot size(\mathcal{T}_2) \leq N$
  possibly label-reduce all $\mathcal{T} \in abs$
  $abs := abs \setminus \{\mathcal{T}_1, \mathcal{T}_2\} \cup \{\mathcal{T}_1 \otimes \mathcal{T}_2\}$
return the remaining abstract transition system in abs

D10.3 Summary

Summary

- There is a wide range of merging strategies. We only covered some important ones.
- Label reduction is crucial for the performance of the merge-and-shrink algorithm, especially when using bisimilarity for shrinking.

D10.4 Literature

Literature (1)

References on merge-and-shrink abstractions:

Klaus Dräger, Bernd Finkbeiner and Andreas Podelski.
Directed Model Checking with Distance-Preserving Abstractions.
Proc. SPIN 2006, pp. 19–34, 2006.
Introduces merge-and-shrink abstractions (for model-checking) and DFP merging strategy.

Malte Helmert, Patrik Haslum and Jörg Hoffmann.
Flexible Abstraction Heuristics for Optimal Sequential Planning.
Proc. ICAPS 2007, pp. 176–183, 2007.
Introduces merge-and-shrink abstractions for planning.

Literature (2)

Raz Nissim, Jörg Hoffmann and Malte Helmert.
Computing Perfect Heuristics in Polynomial Time: On Bisimulation and Merge-and-Shrink Abstractions in Optimal Planning.
Proc. IJCAI 2011, pp. 1983–1990, 2011.
Introduces bisimulation-based shrinking.

Malte Helmert, Patrik Haslum, Jörg Hoffmann and Raz Nissim.
Merge-and-Shrink Abstraction: A Method for Generating Lower Bounds in Factored State Spaces.
Journal of the ACM 61 (3), pp. 16:1–63, 2014.
Detailed journal version of the previous two publications.

Literature (3)

Silvan Sievers, Martin Wehrle and Malte Helmert.
Generalized Label Reduction for Merge-and-Shrink Heuristics.
Proc. AAAI 2014, pp. 2358–2366, 2014.
Introduces label reduction as covered in these slides
(there has been a more complicated version before).

Gaojian Fan, Martin Müller and Robert Holte.
Non-linear merging strategies for merge-and-shrink based on variable interactions.
Proc. AAAI 2014, pp. 2358–2366, 2014.
Introduces UMC and MIASM merging strategies