Latest Trends in Abstraction Heuristics for Classical Planning

3. Merge-and-Shrink Abstractions

Malte Helmert Jendrik Seipp Silvan Sievers

ICAPS 2015 Tutorial

June 7, 2015

Merge-and-Shrink Algorithm

Merge-and-Shrink Abstractions

- Main idea:
  - PDBs: perfectly reflect some state variables
  - M&S: Reflect all state variables in a potentially lossy way
- Theoretically represent arbitrary abstractions of SAS^+ tasks
- Usage as heuristics:
  - Not discussed in this talk
  - Talk at main conference about representational power of merge-and-shrink (Helmert et al., Thursday)

Running Example

Example (one package, two trucks)

Consider the unit-cost SAS$^+$ planning task $\langle V, O, s_0, s_\star \rangle$ with:

- $V = \{p, t_\mathsf{A}, t_\mathsf{B}\}$
- $dom(p) = \{\mathsf{L}, \mathsf{R}, \mathsf{A}, \mathsf{B}\}$ and $dom(t_\mathsf{A}) = dom(t_\mathsf{B}) = \{\mathsf{L}, \mathsf{R}\}$
- $s_0 = \{p \mapsto \mathsf{L}, t_\mathsf{A} \mapsto \mathsf{R}, t_\mathsf{B} \mapsto \mathsf{R}\}$ and $s_\star = \{p \mapsto \mathsf{R}\}$
- $O = \{pickup_{i,j} \mid i \in \{\mathsf{A}, \mathsf{B}\}, j \in \{\mathsf{L}, \mathsf{R}\}\}$
  $\cup \{drop_{i,j} \mid i \in \{\mathsf{A}, \mathsf{B}\}, j \in \{\mathsf{L}, \mathsf{R}\}\}$
  $\cup \{move_{i,j,j'} \mid i \in \{\mathsf{A}, \mathsf{B}\}, j, j' \in \{\mathsf{L}, \mathsf{R}\}, j \neq j'\}$

Running Example

Example (one package, two trucks)

Consider the unit-cost SAS$^+$ planning task $\langle V, O, s_0, s_\star \rangle$ with:

- $V = \{p, t_\mathrm{A}, t_\mathrm{B}\}$
- $dom(p) = \{\mathrm{L}, \mathrm{R}, \mathrm{A}, \mathrm{B}\}$ and $dom(t_\mathrm{A}) = dom(t_\mathrm{B}) = \{\mathrm{L}, \mathrm{R}\}$
- $s_0 = \{p \mapsto \mathrm{L}, t_\mathrm{A} \mapsto \mathrm{R}, t_\mathrm{B} \mapsto \mathrm{R}\}$ and $s_\star = \{p \mapsto \mathrm{R}\}$
- $O = \{pickup_{i,j} \mid i \in \{\mathrm{A}, \mathrm{B}\}, j \in \{\mathrm{L}, \mathrm{R}\}\}$
  $\cup \{drop_{i,j} \mid i \in \{\mathrm{A}, \mathrm{B}\}, j \in \{\mathrm{L}, \mathrm{R}\}\}$
  $\cup \{move_{i,j,j'} \mid i \in \{\mathrm{A}, \mathrm{B}\}, j, j' \in \{\mathrm{L}, \mathrm{R}\}, j \neq j'\}$

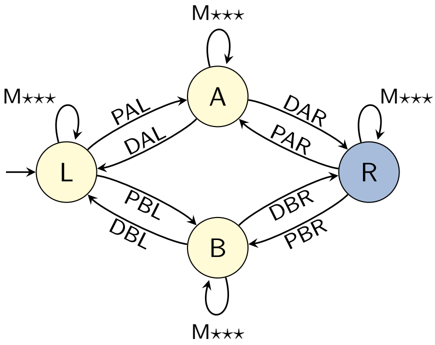
- Abbreviations for labels:
  - MALR: move truck A from left to right
  - DAR: drop package from truck A at right location
  - $\star$: wildcard

Initialization

- Maintain a set X of current TS
- Initialize X to contain all TS corresponding to atomic projections

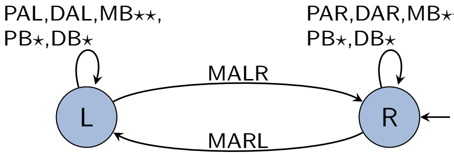
Running Example: Atomic Projections

$\Theta^{\pi_{\{package\}}}$:



Running Example: Atomic Projections

$\Theta^{\pi_{\{truck\ A\}}}$:



Synchronized product

Definition (synchronized product)

For $i \in \{1, 2\}$, let $\Theta^i = \langle S^i, L, T^i, s_0^i, S_\star^i \rangle$ be TS (with identical labels and cost)

The synchronized product of Θ^1 and Θ^2, in symbols $\Theta^1 \otimes \Theta^2$, is the TS $\Theta^\otimes = \langle S^\otimes, L, T^\otimes, s_0^\otimes, S_\star^\otimes \rangle$ with

- $S^\otimes := S^1 \times S^2$
- $T^\otimes := \{\langle\langle s_1, s_2\rangle, \ell, \langle t_1, t_2\rangle\rangle \mid \langle s_1, \ell, t_1\rangle \in T^1 \wedge \langle s_2, \ell, t_2\rangle \in T^2\}$
- $s_0^\otimes := \langle s_0^1, s_0^2 \rangle$
- $S_\star^\otimes := S_\star^1 \times S_\star^2$

Synchronized product

Definition (synchronized product)

For $i \in \{1, 2\}$, let $\Theta^i = \langle S^i, L, T^i, s_0^i, S_\star^i \rangle$ be TS (with identical labels and cost)

The synchronized product of Θ^1 and Θ^2, in symbols $\Theta^1 \otimes \Theta^2$, is the TS $\Theta^\otimes = \langle S^\otimes, L, T^\otimes, s_0^\otimes, S_\star^\otimes \rangle$ with

- $S^\otimes := S^1 \times S^2$
- $T^\otimes := \{\langle \langle s_1, s_2 \rangle, \ell, \langle t_1, t_2 \rangle \rangle \mid \langle s_1, \ell, t_1 \rangle \in T^1 \wedge \langle s_2, \ell, t_2 \rangle \in T^2\}$
- $s_0^\otimes := \langle s_0^1, s_0^2 \rangle$
- $S_\star^\otimes := S_\star^1 \times S_\star^2$

Theorem

Let Π *be an* SAS$^+$ *planning task with variables set* V*.*
Then $\Theta(\Pi) \sim \bigotimes_{v \in V} \Theta^{\pi\{v\}}$*.*

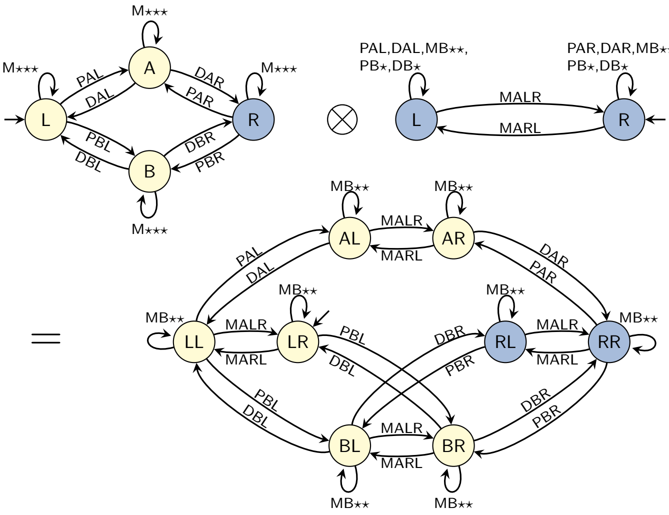
Merging

Definition (merging)

Merging is the operation that replaces two TS by their synchronized product in the current set of TS X.

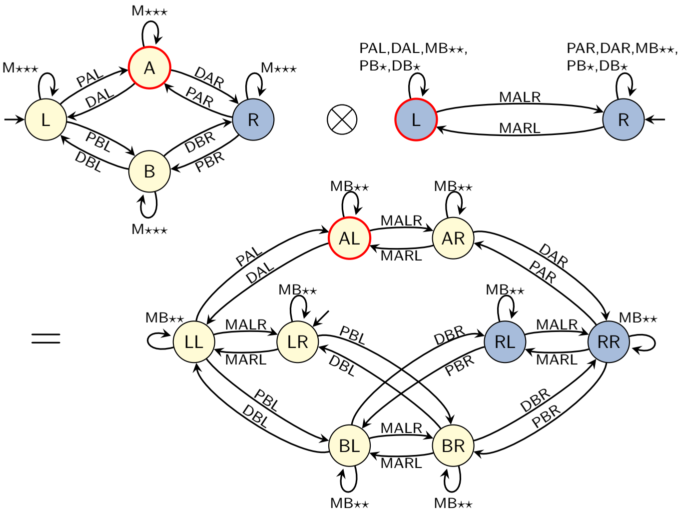
Running Example: Merging

$\Theta^1 := \Theta^{\pi\{package\}} \otimes \Theta^{\pi\{truck\ A\}}$:



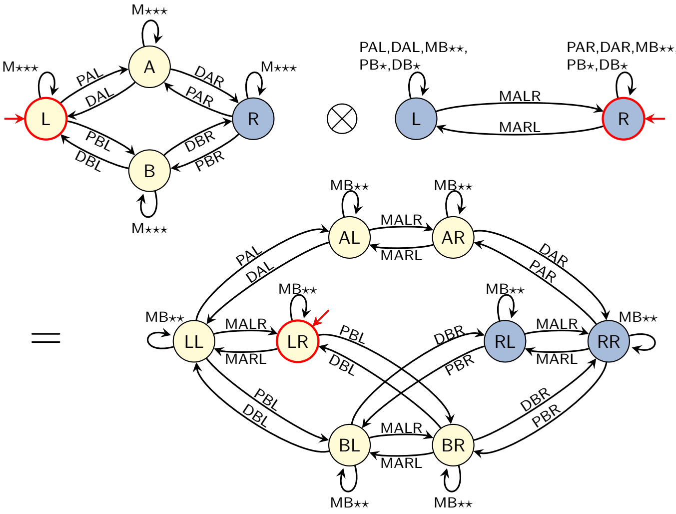
Running Example: Merging

$\Theta^1 := \Theta^{\pi\{package\}} \otimes \Theta^{\pi\{truck\ A\}}$: $S^{\otimes} = S^1 \times S^2$



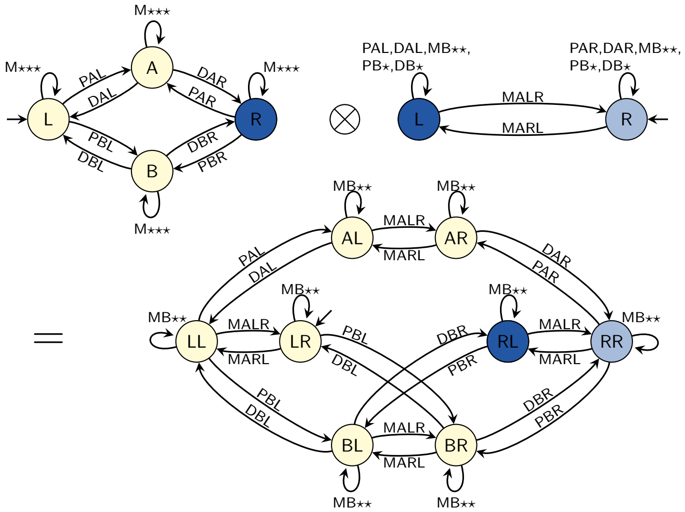
Running Example: Merging

$\Theta^1 := \Theta^{\pi\{package\}} \otimes \Theta^{\pi\{truck\ A\}}$: $s_0^{\otimes} = \langle s_0^1, s_0^2 \rangle$



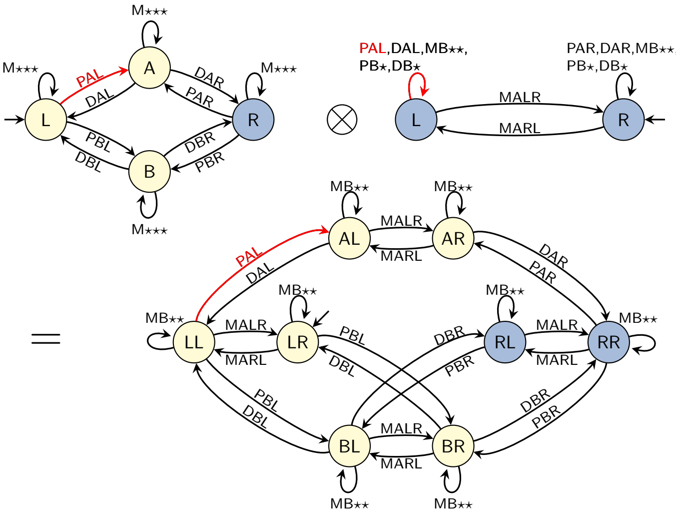
Running Example: Merging

$\Theta^1 := \Theta^{\pi\{package\}} \otimes \Theta^{\pi\{truck\ A\}}$: $S^{\otimes}_{\star} = S^1_{\star} \times S^2_{\star}$



Running Example: Merging

$\Theta^1 := \Theta^{\pi\{package\}} \otimes \Theta^{\pi\{truck\ A\}}$: $T^{\otimes} = \{\langle\langle s_1, s_2\rangle, a, \langle t_1, t_2\rangle\rangle \mid \ldots\}$



Merging

Theorem

Merging is an exact operation, i.e. it preserves goal states, transitions, and label cost.

Shrinking

Definition (abstraction)

An abstraction of a transition system Θ with states S is a function $\alpha : S \to S'$.

Shrinking

Definition (abstraction)

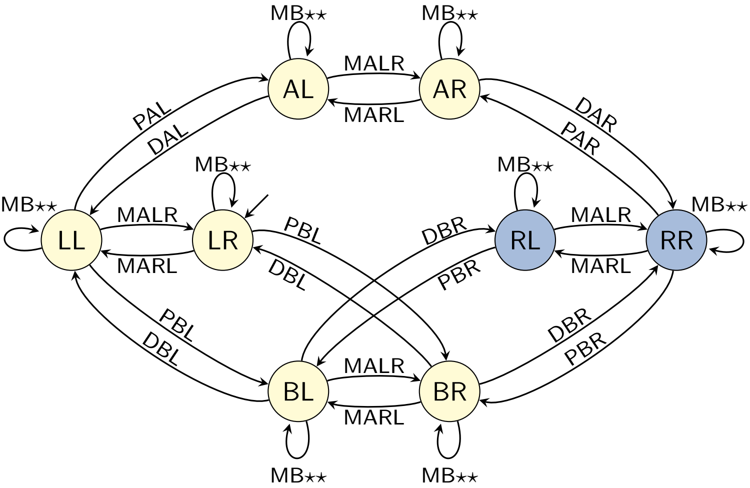
An abstraction of a transition system Θ with states S is a function $\alpha : S \to S'$.

Definition (shrinking)

Shrinking is the operation that applies an abstraction to a TS in the current set of TS X.

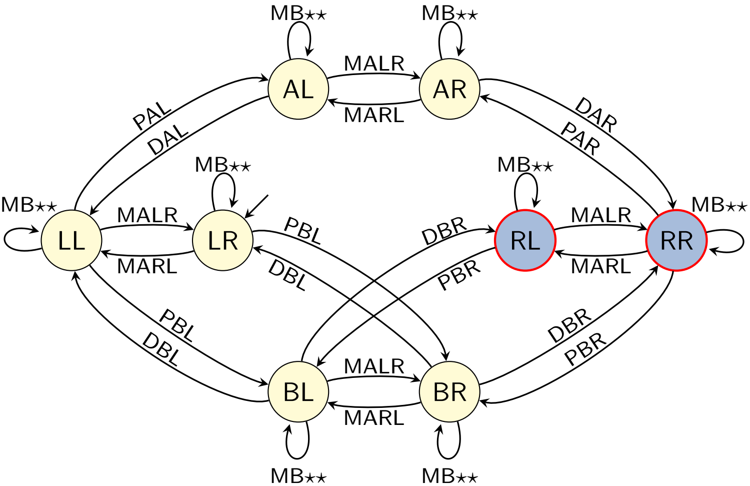
Running Example: Shrinking

$\Theta^2 :=$ an abstraction of Θ^1



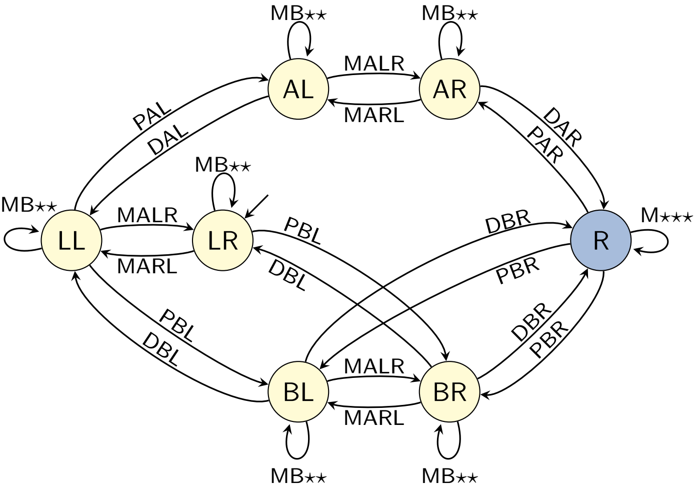
Running Example: Shrinking

$\Theta^2 :=$ an abstraction of Θ^1



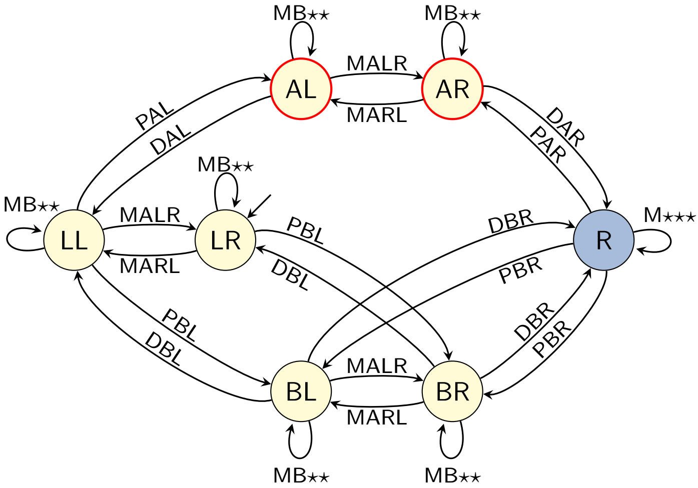
Running Example: Shrinking

$\Theta^2 :=$ an abstraction of Θ^1



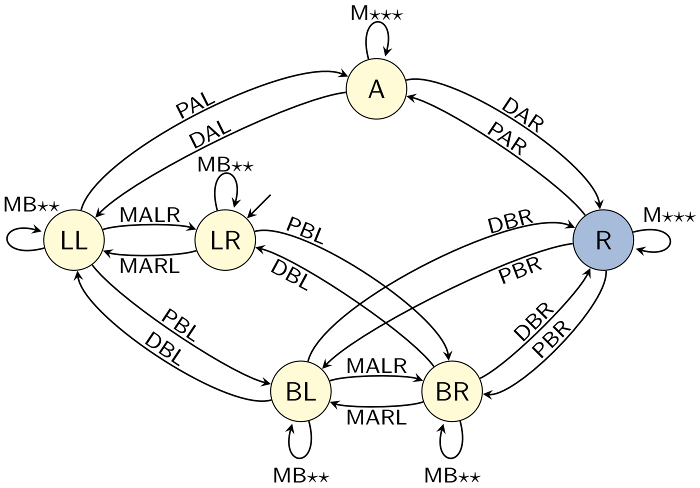
Running Example: Shrinking

$\Theta^2 :=$ an abstraction of Θ^1



Running Example: Shrinking

$\Theta^2 :=$ an abstraction of Θ^1



Shrinking

Theorem

Shrinking is a safe operation, i.e. it preserves goal states, does not increase label cost, and all concrete transitions have a corresponding abstract transition.

Label Reduction

Definition (generalized label reduction)

Let L be the common label set of all TS in the current set of TS X. A label reduction is a mapping τ defined on L which satisfies $cost(\tau(\ell)) \leq cost(\ell)$ for all $\ell \in L$.

Label Reduction

Definition (generalized label reduction)

Let L be the common label set of all TS in the current set of TS X. A label reduction is a mapping τ defined on L which satisfies $cost(\tau(\ell)) \leq cost(\ell)$ for all $\ell \in L$.

Theorem

Label reduction is safe.

Exact Label Reduction

Definitions

- Labels are locally equivalent in TS Θ if they label the same set of transitions in Θ.
- Labels are Θ-combinable for TS Θ if they are locally equivalent in all TS in X but Θ.

Exact Label Reduction

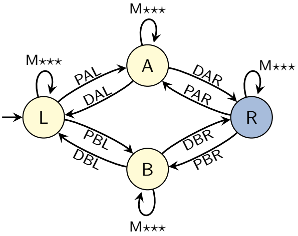
Definitions

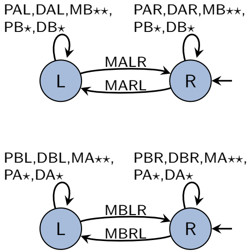
- Labels are locally equivalent in TS Θ if they label the same set of transitions in Θ.
- Labels are Θ-combinable for TS Θ if they are locally equivalent in all TS in X but Θ.

Theorem

A label reduction τ is exact if it only combines labels of the same cost that are Θ-combinable for some $\Theta \in X$.

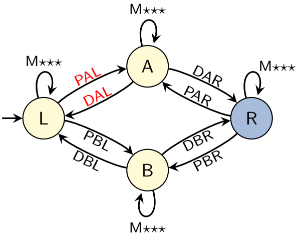
Running Example: Exact Label Reduction

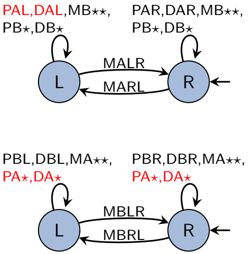




Running Example: Exact Label Reduction

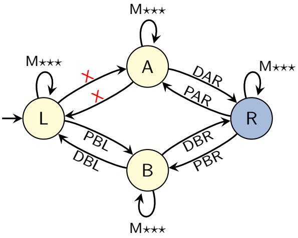
- Exact label reduction: PAL, DAL

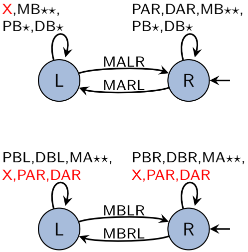




Running Example: Exact Label Reduction

- Exact label reduction: $\tau(\text{PAL}) = \text{X}, \tau(\text{DAL}) = \text{X}$





Merge-and-Shrink Algorithm

Generic algorithm to compute M&S abstractions

$X := \{\Theta^{\pi_{\{v\}}} \mid v \in V\}$ [TS for atomic projections]

while $|X| > 1$:

  apply exact label reduction to L (and all TS in X)

  select Θ^1, Θ^2 from X

  shrink Θ^1 and/or Θ^2 until $\text{size}(\Theta^1) \cdot \text{size}(\Theta^2) \leq K$

  $X := X \setminus \{\Theta^1, \Theta^2\} \cup \{\Theta^1 \otimes \Theta^2\}$ [Merging]

return the remaining TS in X, an abstraction of Π

Merge-and-Shrink Algorithm

Generic algorithm to compute M&S abstractions

$X := \{\Theta^{\pi_{\{v\}}} \mid v \in V\}$ [TS for atomic projections]
while $|X| > 1$:
    apply exact label reduction to L (and all TS in X)
    select Θ^1, Θ^2 from X
    shrink Θ^1 and/or Θ^2 until $\text{size}(\Theta^1) \cdot \text{size}(\Theta^2) \leq K$
    $X := X \setminus \{\Theta^1, \Theta^2\} \cup \{\Theta^1 \otimes \Theta^2\}$ [Merging]
return the remaining TS in X, an abstraction of Π

- How to choose K? $\rightsquigarrow$ according to memory constraints
- How to select Θ^1, Θ^2? $\rightsquigarrow$ Merging strategies
- How to shrink? $\rightsquigarrow$ Shrinking strategies

Shrinking Strategies

F-preserving

- Repeatedly combine state with the same g- and h-values if possible
- Prefer high f-values (high h-values)
- Lose precision only in regions far away from goal states

Bisimulation

Definition (bisimulation)

Let $\Theta = \langle S, L, T, s_0, S_\star \rangle$ be a TS. An equivalence relation $\sim$ on S is a bisimulation for Θ if $s \sim t$ implies that

1. either $s, t \in S_\star$ or $s, t \notin S_\star$, and
2. for all $\langle s, \ell, s' \rangle \in T$ there exists $\langle t, \ell, t' \rangle \in T$ and $s' \sim t'$.

Bisimulation

Definition (bisimulation)

Let $\Theta = \langle S, L, T, s_0, S_\star \rangle$ be a TS. An equivalence relation $\sim$ on S is a bisimulation for Θ if $s \sim t$ implies that

1. either $s, t \in S_\star$ or $s, t \notin S_\star$, and
2. for all $\langle s, \ell, s' \rangle \in T$ there exists $\langle t, \ell, t' \rangle \in T$ and $s' \sim t'$.

- Shrinking: combine bisimilar states
- Exact if based on full bisimulation
- In practice: bisimulation too large, approximate

Merging Strategies

Merging Strategies

- Linear merging strategies:
  - Based on a (precomputed) order of variables
  - Only maintain one large TS at a time

Merging Strategies

- Linear merging strategies:
  - Based on a (precomputed) order of variables
  - Only maintain one large TS at a time
- Non-linear merging strategy: DFP
  - Preferably merge TS with labels inducing transitions close to goal states
  - Build fine-grained abstractions around goal states

Merging Strategies

- Linear merging strategies:
  - Based on a (precomputed) order of variables
  - Only maintain one large TS at a time
- Non-linear merging strategy: DFP
  - Preferably merge TS with labels inducing transitions close to goal states
  - Build fine-grained abstractions around goal states
- Non-linear merging strategy: MIASM
  - Preferably merge TS such that the product contains many unreachable and irrelevant states
  - Prune TS by removing unreachable and irrelevant states

Non-Linear Merging Based on Symmetries

- Factored symmetries:
  - Permutation of states and labels of all TS
  - Preserve structure and path cost of all TS
  - Local if states not mapped accross different TS

How to Use Local Factored Symmetries

- Merging based on local factored symmetries:
  - Compute a symmetry affecting several TS
  - Merge all affected TS, the symmetry is then atomic, i.e. affects only one TS

How to Use Local Factored Symmetries

- Merging based on local factored symmetries:
  - Compute a symmetry affecting several TS
  - Merge all affected TS, the symmetry is then atomic, i.e. affects only one TS
- Why are atomic symmetries useful?
  - Shrinking based on atomic symmetries is exact
  - Atomic symmetries implicitly captured by shrinking based on bisimulation and label reduction

Literature

- Dräger et al., SPIN 2006: original contribution (model checking)
- Helmert et al., ICAPS 2007: adaptation to planning (f-preserving shrinking, linear merging)
- Nissim et al., IJCAI 2011: bisimulation based shrinking
- Edelkamp et al., ECAI 2012: symbolic m&s
- Torralba et al., IJCAI 2013: symbolic m&s
- Helmert et al., JACM 2014: overview journal paper
- Sievers et al., AAAI 2014: generalized label reduction, DFP
- Fan and Holte, SoCS 2014: MIASM
- Hoffmann et al., ECAI 2014: m&s for detecting unsolvability
- Sievers et al., AAAI 2015: symmetry based merging

Pointers for the Conference

- Torralba and Hoffmann, IJCAI 2015 (Monday in the HSDIP workshop):

  Simulation-Based Admissible Dominance Pruning

- Helmert et al., ICAPS 2015 (Tuesday):

  On the Expressive Power of
  Non-Linear Merge-and-Shrink Representations

- Torralba and Kissmann, SoCS 2015 (Thursday in the joint ICAPS/SoCS session):

  Focusing on What Really Matters:
  Irrelevance Pruning in Merge-and-Shrink