Planning and Optimization

D7. Merge-and-Shrink Abstractions: Synchronized Product

Malte Helmert and Gabriele Röger

Universität Basel

November 13, 2017

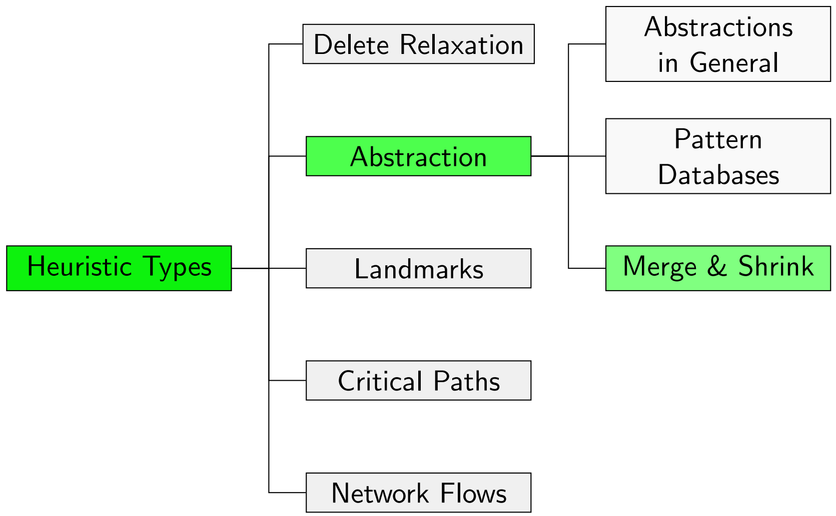
D7.1 Motivation

D7.2 Synchronized Product

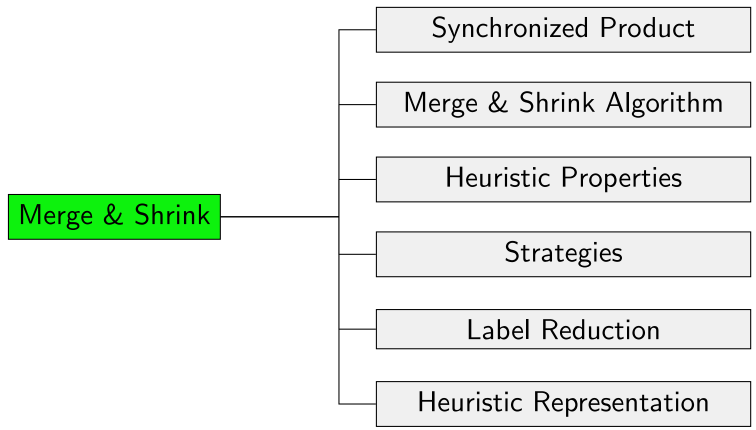
D7.3 Synchronized Products and Abstractions

D7.4 Summary

Content of this Course: Heuristic Types



Content of this Course: Merge & Shrink

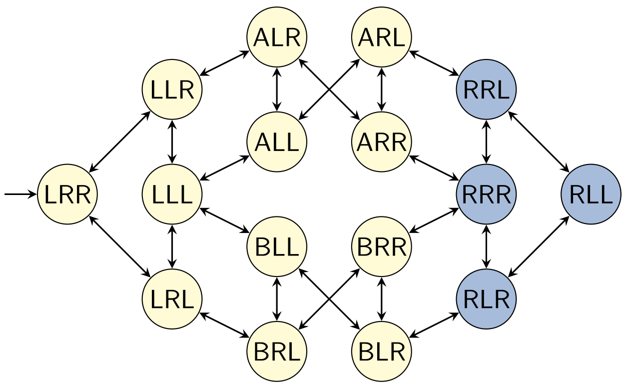


D7.1 Motivation

Beyond Pattern Databases

- Despite their popularity, pattern databases have some fundamental limitations ($\rightsquigarrow$ example on next slides).
- This week, we study a class of abstractions called merge-and-shrink abstractions.
- Merge-and-shrink abstractions can be seen as a proper generalization of pattern databases.
  - They can do everything that pattern databases can do (modulo polynomial extra effort).
  - They can do some things that pattern databases cannot.

Back to the Running Example

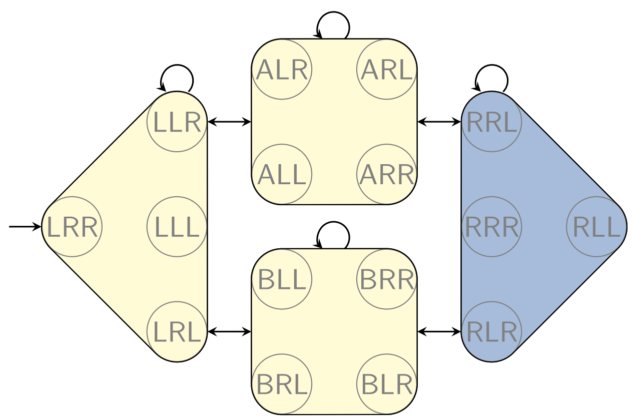


Logistics problem with one package, two trucks, two locations:

- state variable package: $\{L, R, A, B\}$
- state variable truck A: $\{L, R\}$
- state variable truck B: $\{L, R\}$

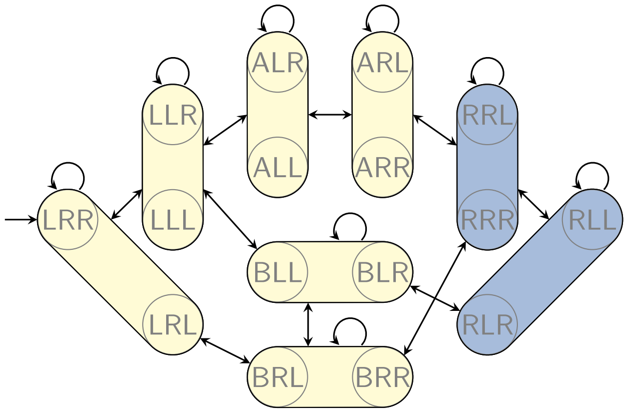
Example: Projection

$\mathcal{T}^{\pi_{\{\text{package}\}}}$:



Example: Projection (2)

$\mathcal{T}^{\pi_{\{\text{package,truck A}\}}}$:



Limitations of Projections

How accurate is the PDB heuristic?

- consider generalization of the example: N trucks, M locations (fully connected), still one package
- consider any pattern that is a proper subset of variable set V.
- $h(s_0) \leq 2 \rightsquigarrow$ no better than atomic projection to package

These values cannot be improved by maximizing over several patterns or using additive patterns.

Merge-and-shrink abstractions can represent heuristics with $h(s_0) \geq 3$ for tasks of this kind of any size.
Time and space requirements are polynomial in N and M.

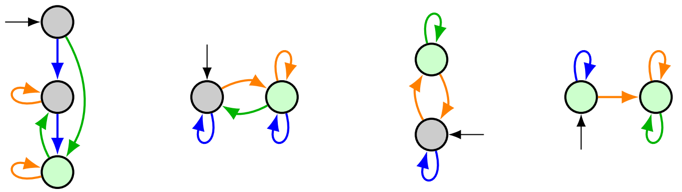
Merge-and-Shrink Abstractions: Main Idea

Main Idea of Merge-and-shrink Abstractions
(due to Dräger, Finkbeiner & Podelski, 2006):

Instead of perfectly reflecting a few state variables, reflect all state variables, but in a potentially lossy way.

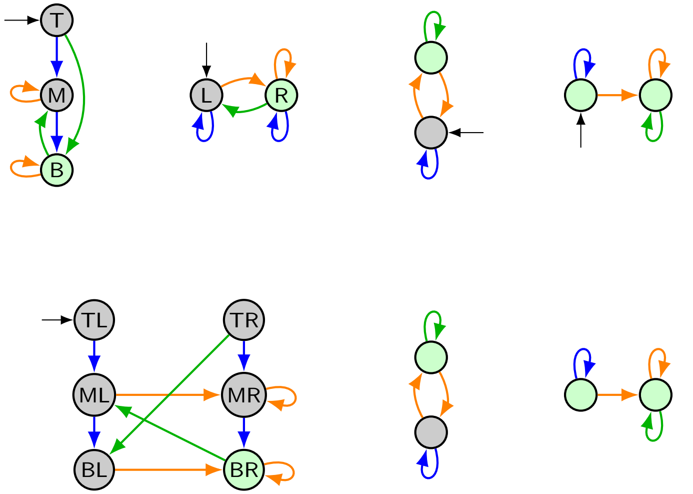
Merge-and-Shrink Abstractions: Idea

Start from projections to single state variables



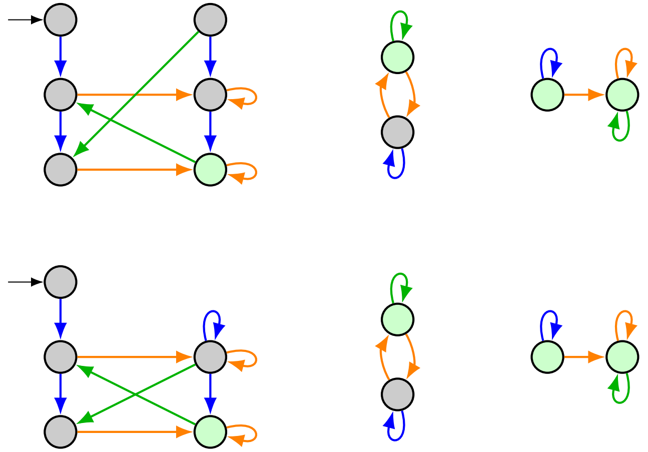
Merge-and-Shrink Abstractions: Idea

Successively replace two transition systems with their product.



Merge-and-Shrink Abstractions: Idea

If too large, replace a transition system with an abstract system.



Merge-and-Shrink Abstractions: Idea

- Given two abstract transition systems, we can merge them into a new abstract product transition system.
- The product transition system captures all information of both transition systems and can be better informed than either.
- It can even be better informed than their sum.
- If merging with another abstract transition system exceeded memory limitations, we can shrink an intermediate result using any abstraction and then continue the merging process.

D7.2 Synchronized Product

Synchronized Product

Merge & Shrink

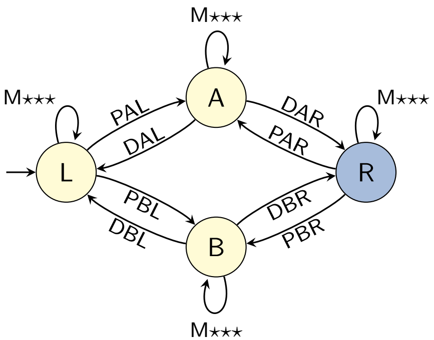
- **Synchronized Product**
- Merge & Shrink Algorithm
- Heuristic Properties
- Strategies
- Label Reduction
- Heuristic Representation

Running Example: Explanations

- Atomic projections – projections to a single state variable – play an important role for merge-and-shrink abstractions.
- Unlike previous chapters, transition labels are critically important for this topic.
- Hence we now look at the transition systems for atomic projections of our example task, including transition labels.
- We abbreviate operator names as in these examples:
  - MALR: move truck A from left to right
  - DAR: drop package from truck A at right location
  - PBL: pick up package with truck B at left location
- We abbreviate parallel arcs with commas and wildcards ($\star$) in the labels as in these examples:
  - PAL, DAL: two parallel arcs labeled PAL and DAL
  - MA$\star\star$: two parallel arcs labeled MALR and MARL

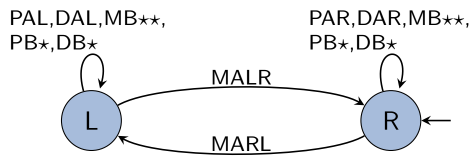
Running Example: Atomic Projection for Package

$\mathcal{T}^{\pi_{\{\text{package}\}}}$:



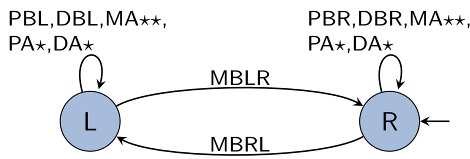
Running Example: Atomic Projection for Truck A

$\mathcal{T}^{\pi_{\{\text{truck A}\}}}$:



Running Example: Atomic Projection for Truck B

$\mathcal{T}^{\pi_{\{\text{truck B}\}}}$:



Synchronized Product of Transition Systems

Definition (Synchronized Product of Transition Systems)

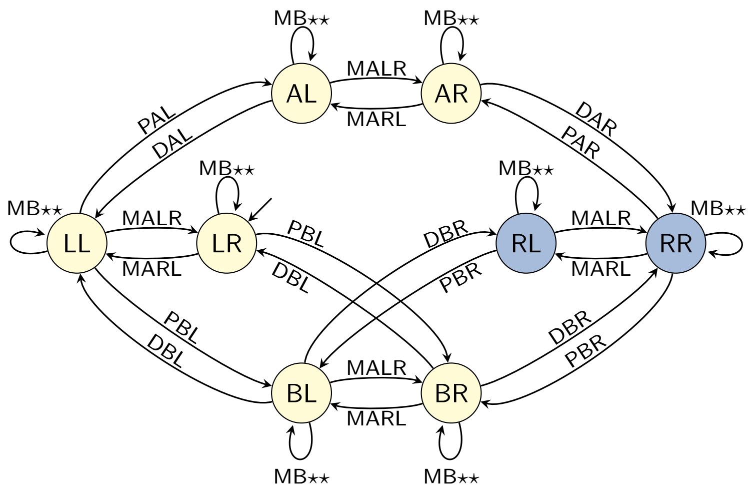
For $i \in \{1, 2\}$, let $\mathcal{T}_i = \langle S_i, L, c, T_i, s_{0i}, S_{\star i}\rangle$ be transition systems with identical label set and identical label cost function.

The synchronized product of $\mathcal{T}_1$ and $\mathcal{T}_2$, in symbols $\mathcal{T}_1 \otimes \mathcal{T}_2$, is the transition system $\mathcal{T}_\otimes = \langle S_\otimes, L, c, T_\otimes, s_{0\otimes}, S_{\star\otimes}\rangle$ with

- $S_\otimes := S_1 \times S_2$
- $T_\otimes := \{\langle\langle s_1, s_2\rangle, l, \langle t_1, t_2\rangle\rangle \mid \langle s_1, l, t_1\rangle \in T_1 \text{ and } \langle s_2, l, t_2\rangle \in T_2\}$
- $s_{0\otimes} := \langle s_{01}, s_{02}\rangle$
- $S_{\star\otimes} := S_{\star 1} \times S_{\star 2}$

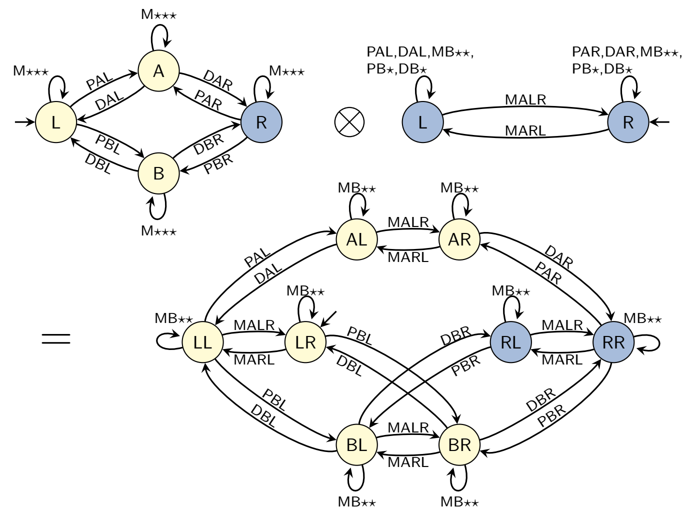
Example: Synchronized Product

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}}$:



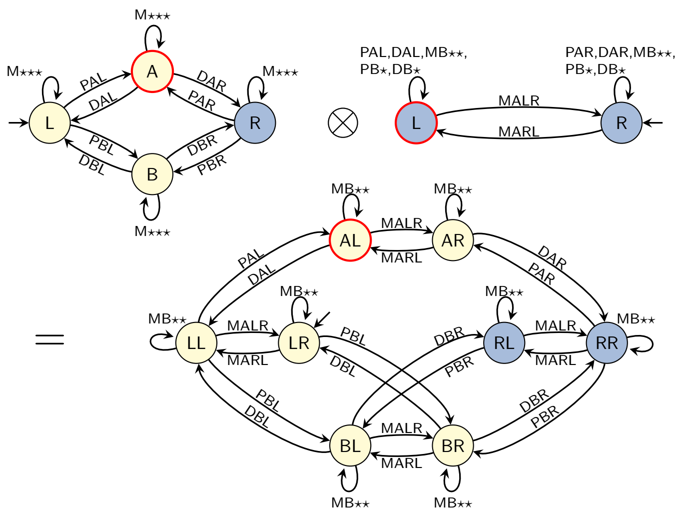
Example: Computation of Synchronized Product

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}}$:



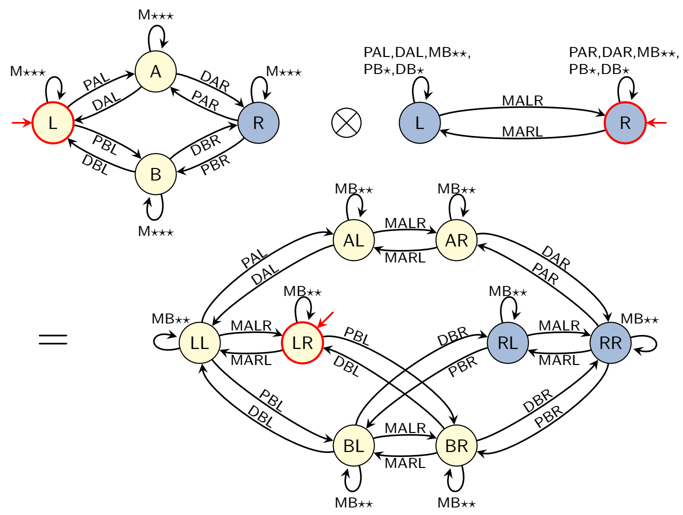
Example: Computation of Synchronized Product

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}}$: $S_\otimes = S_1 \times S_2$



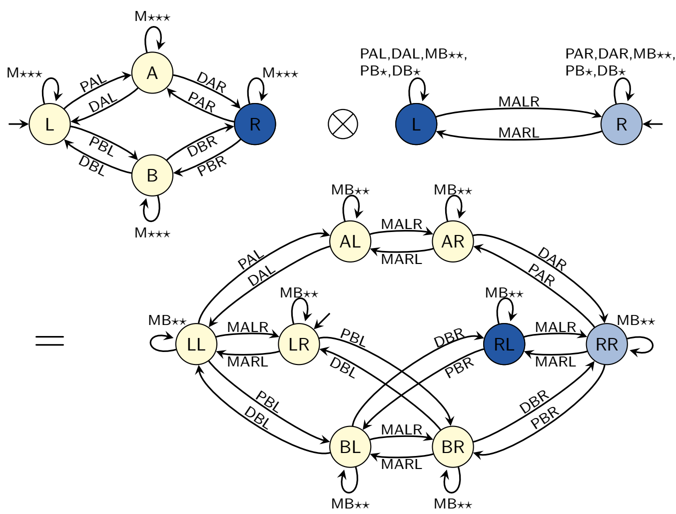
Example: Computation of Synchronized Product

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}}$: $s_{0\otimes} = \langle s_{01}, s_{02} \rangle$



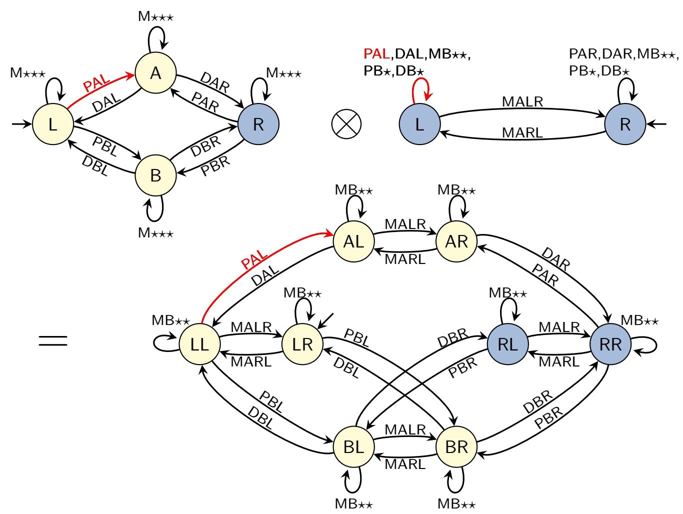
Example: Computation of Synchronized Product

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}}$: $S_{\star\otimes} = S_{\star 1} \times S_{\star 2}$



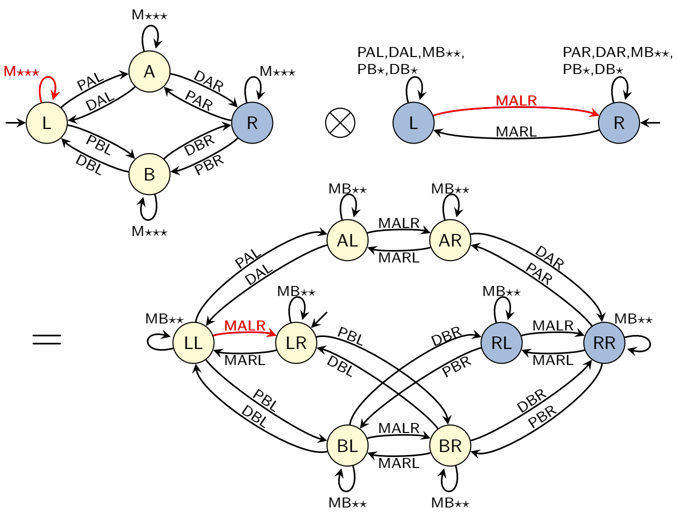
Example: Computation of Synchronized Product

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}}$: $T_\otimes := \{\langle\langle s_1, s_2\rangle, l, \langle t_1, t_2\rangle\rangle \mid \dots\}$



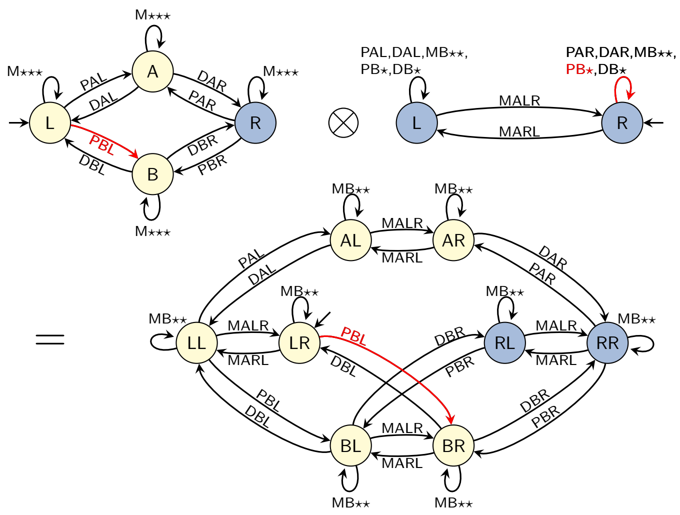
Example: Computation of Synchronized Product

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}}$: $T_\otimes := \{\langle\langle s_1, s_2\rangle, l, \langle t_1, t_2\rangle\rangle \mid \dots\}$



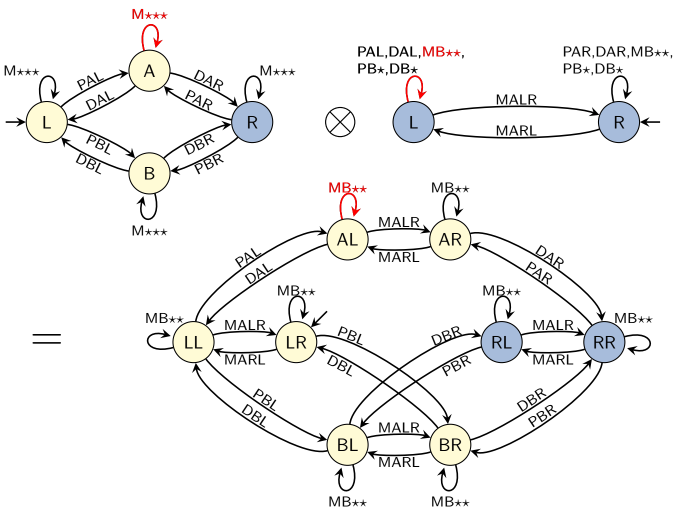
Example: Computation of Synchronized Product

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}}$: $T_\otimes := \{\langle\langle s_1, s_2\rangle, l, \langle t_1, t_2\rangle\rangle \mid \dots\}$



Example: Computation of Synchronized Product

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}}$: $T_\otimes := \{\langle\langle s_1, s_2\rangle, l, \langle t_1, t_2\rangle\rangle \mid \dots\}$



D7.3 Synchronized Products and Abstractions

Synchronized Product of Functions

Definition (Synchronized Product of Functions)

Let $\alpha_1 : S \to S_1$ and $\alpha_2 : S \to S_2$ be functions with identical domain.

The synchronized product of α_1 and α_2, in symbols $\alpha_1 \otimes \alpha_2$, is the function $\alpha_\otimes : S \to S_1 \times S_2$ defined as $\alpha_\otimes(s) = \langle \alpha_1(s), \alpha_2(s) \rangle$.

Synchronized Product of Abstractions

Theorem

Let α_1 and α_2 be abstractions of transition system $\mathcal{T}$ such that $\alpha_\otimes := \alpha_1 \otimes \alpha_2$ is surjective.

Then $\alpha_\otimes$ is an abstraction of $\mathcal{T}$ and a refinement of α_1 and α_2.

Proof.

Abstraction: suitable domain as α_1, α_2 are abstractions of $\mathcal{T}$, surjective by premise

Refinement: For $i \in \{1, 2\}$, $\alpha_i = \beta_i \circ \alpha_\otimes$ with $\beta_i(\langle x_1, x_2 \rangle) = x_i$. □

Preserving Abstractions

- It would be very nice if we could prove that if α_1 and α_2 are abstractions of $\mathcal{T}$ then there is an abstraction of $\mathcal{T}$ inducing $\mathcal{T}^{\alpha_1} \otimes \mathcal{T}^{\alpha_2}$.
- However, this is not true in general.
- It is not even true for SAS$^+$ tasks.
- But there is an important sufficient condition for preserving the abstraction property.

Synchronized Products and Abstractions

Theorem (Synchronized Products and Abstractions)

Let Π be a SAS$^+$ planning task with variable set V, and let V_1 and V_2 be disjoint subsets of V.

For $i \in \{1, 2\}$, let α_i be an abstraction of $\mathcal{T}(\Pi)$ such that α_i is a coarsening of π_{V_i}.

Then $\alpha_\otimes := \alpha_1 \otimes \alpha_2$ is surjective and $\mathcal{T}^{\alpha_1 \otimes \alpha_2} = \mathcal{T}^{\alpha_1} \otimes \mathcal{T}^{\alpha_2}$.

Synchronized Products and Abstractions

Proof.

Let $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$ and
for $i \in \{1, 2\}$ let $\mathcal{T}^{\alpha_i} = \langle S_i, L, c, T_i, s_{0i}, S_{\star i} \rangle$ (with $\alpha_i : S \to S_i$).

$\alpha_1 \otimes \alpha_2$ is surjective:

Since α_i is a coarsening of π_{V_i} there is a β_i such that $\alpha_i = \beta_i \circ \pi_{V_i}$ with $\beta_i : S|_{V_i} \to S_i$.
Consider an arbitrary $\langle s_1, s_2 \rangle \in S_1 \times S_2$.
As α_1, α_2 are surjective (because they are abstractions), there are $s'_1, s'_2 \in S$ such that $\alpha_i(s'_i) = s_i$.
As S consists of all valuations of V, also state s with $s|_{V_1} = s'_1|_{V_1}$ and $s|_{V \setminus V_1} = s'_2|_{V \setminus V_1}$ is in S.
Then $\alpha_i(s) = \beta_i \circ \pi_{V_i}(s) = \beta_i \circ \pi_{V_i}(s'_i) = \alpha_i(s'_i) = s_i$ and hence $\alpha_1 \otimes \alpha_2(s) = \langle \alpha_1(s), \alpha_2(s) \rangle = \langle s_1, s_2 \rangle$. ...

Synchronized Products and Abstractions

Proof (continued).

$\mathcal{T}^{\alpha_1 \otimes \alpha_2} = \mathcal{T}^{\alpha_1} \otimes \mathcal{T}^{\alpha_2}$:

$$\begin{aligned}
S_{\alpha_1 \otimes \alpha_2} &= S_1 \times S_2 = S_\otimes \\
s_{0\alpha_1 \otimes \alpha_2} &= \alpha_1 \otimes \alpha_2(s_0) = \langle \alpha_1(s_0), \alpha_2(s_0) \rangle = \langle s_{01}, s_{02} \rangle = s_{0\otimes} \\
S_{\star \alpha_1 \otimes \alpha_2} &= \{\alpha_1 \otimes \alpha_2(s) \mid s \in S_\star\} \\
&= \{\langle \alpha_1(s), \alpha_2(s) \rangle \mid s \in S_\star\} \\
&\subseteq \{\langle \alpha_1(s), \alpha_2(s') \rangle \mid s, s' \in S_\star\} \\
&= \{\langle s_1, s_2 \rangle \mid s_1 \in S_{\star 1}, s_2 \in S_{\star 2}\} \\
&= S_{\star 1} \times S_{\star 2} \\
&= S_{\star \otimes}
\end{aligned}$$

...

Synchronized Products and Abstractions

Proof (continued).

For equality, we also need to establish that
$\{\langle \alpha_1(s), \alpha_2(s') \rangle \mid s, s' \in S_\star\} \subseteq \{\langle \alpha_1(s), \alpha_2(s) \rangle \mid s \in S_\star\}$.

Consider arbitrary $s, s' \in S_\star$.
Define s'' as $s''|_{V_1} = s|_{V_1}$ and $s''|_{V \setminus V_1} = s'|_{V \setminus V_1}$.

It holds that $\alpha_1(s'') = \alpha_1(s)$ and $\alpha_2(s'') = \alpha_2(s')$ because α_i is a coarsening of π_{V_i}.

Furthermore, $s'' \in S_\star$: the goal formula γ of a SAS$^+$ task is a conjunction of atoms $v = d$. If $v \in V_1$, then $s''(v) = d$ because $s \in S_\star$, otherwise $s''(v) = d$ because $s' \in S_\star$. Overall, $s'' \models \gamma$. ...

Synchronized Products and Abstractions

Proof (continued).

We still need to show the equality of the sets of transitions.

$$\begin{aligned}
T_{\alpha_1 \otimes \alpha_2} &= \{\langle \alpha_1 \otimes \alpha_2(s), o, \alpha_1 \otimes \alpha_2(t) \rangle \mid \langle s, o, t \rangle \in T\} \\
&= \{\langle \langle \alpha_1(s), \alpha_2(s) \rangle, o, \langle \alpha_1(t), \alpha_2(t) \rangle \rangle \mid \langle s, o, t \rangle \in T\} \\
&\subseteq \{\langle \langle \alpha_1(s), \alpha_2(s') \rangle, o, \langle \alpha_1(t), \alpha_2(t') \rangle \rangle \\
&\qquad \mid \langle s, o, t \rangle, \langle s', o, t' \rangle \in T\} \\
&= \{\langle \langle s_1, s_2 \rangle, o, \langle t_1, t_2 \rangle \rangle \mid \langle s_1, o, t_1 \rangle \in T_1, \langle s_2, o, t_2 \rangle \in T_2\} \\
&= T_\otimes
\end{aligned}$$

For equality, we need to show that for $\langle s, o, t \rangle, \langle s', o, t' \rangle \in T$ there is a transition $\langle s'', o, t'' \rangle \in T$ with
$\alpha_1(s) = \alpha_1(s''), \alpha_1(t) = \alpha_1(t''), \alpha_2(s') = \alpha_2(s''), \alpha_2(t') = \alpha_2(t'')$. ...

Synchronized Products and Abstractions

Proof (continued).

Consider $s'' \in S$ with $s''|_{V_1} = s|_{V_1}$ and $s''|_{V \setminus V_1} = s'|_{V \setminus V_1}$ and $t'' \in S$ with $t''|_{V_1} = t|_{V_1}$ and $t''|_{V \setminus V_1} = t'|_{V \setminus V_1}$.

Since $pre(o)$ is a conjunction of atoms and $consist(eff(o)) \equiv \top$, o is applicable in s'' by an analogous argument as for the goal.

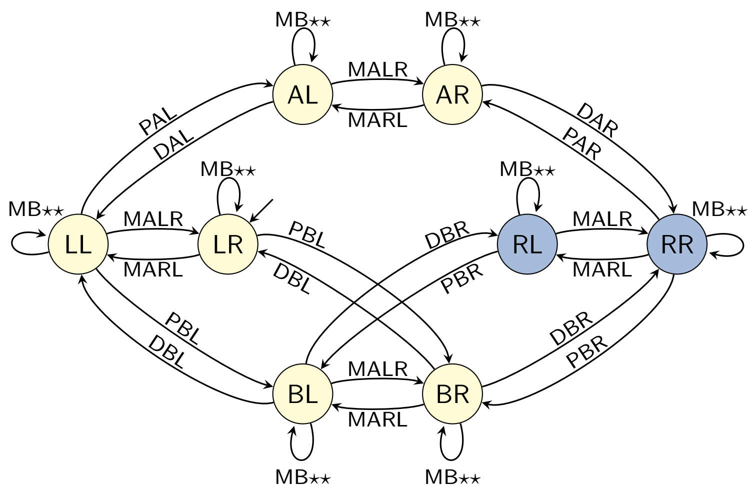
As $t = s[\![o]\!]$, we have $t|_{V \setminus vars(eff(o))} = s|_{V \setminus vars(eff(o))}$, analogously for t' and s'. Hence $t''|_{V \setminus vars(eff(o))} = s''|_{V \setminus vars(eff(o))}$.

As $eff(o)$ contains no conditional effect, it holds for all atomic effects $v := d$ in $eff(o)$ that $t(v) = t'(v) = d$ and hence $t''(v) = d$. Overall, $t'' = s''[\![o]\!]$ and $\langle s'', \ell, t'' \rangle \in T$.

The requirements on the abstractions are again satisfied by the construction of s'' and t'' and α_i being coarsenings of π_{V_i}. □

Example: Product for Disjoint Projections

$\mathcal{T}^{\pi_{\{\text{package}\}}} \otimes \mathcal{T}^{\pi_{\{\text{truck A}\}}} \sim \mathcal{T}^{\pi_{\{\text{package},\text{truck A}\}}}$:



Synchronized Products of Projections

Corollary (Synchronized Products of Projections)

Let Π be a SAS$^+$ planning task with variable set V, and let V_1 and V_2 be disjoint subsets of V. Then $\mathcal{T}^{\pi_{V_1}} \otimes \mathcal{T}^{\pi_{V_2}} \sim \mathcal{T}^{\pi_{V_1 \cup V_2}}$.

(Proof omitted.)

By repeated application of the corollary, we can recover all pattern database heuristics of a SAS$^+$ planning task from the abstract transition systems induced by atomic projections.

Recovering $\mathcal{T}(\Pi)$ from the Atomic Projections

Moreover, by computing the product of all atomic projections, we can recover the identity abstraction id $= \pi_V$.

Corollary (Recovering $\mathcal{T}(\Pi)$ from the Atomic Projections)

Let Π be a SAS$^+$ planning task with variable set V. Then $\mathcal{T}(\Pi) \sim \bigotimes_{v \in V} \mathcal{T}^{\pi_{\{v\}}}$.

This is an important result because it shows that the transition systems induced by atomic projections contain all information of a SAS$^+$ task.

D7.4 Summary

Summary

- The synchronized product of two transition systems captures "what we can do" in both systems "in parallel".
- With suitable abstractions, the synchronized product of the induced transition systems is induced by the synchronized product of the abstractions.
- We can recover the original transition system from the abstract transition systems induced by the atomic projections.