Planning and Optimization

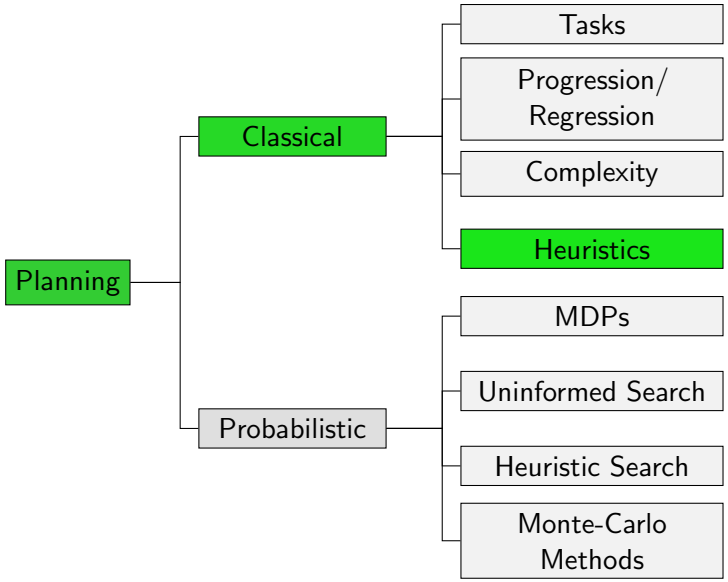
D7. M&S: Generic Algorithm and Heuristic Properties

Gabriele Röger and Thomas Keller

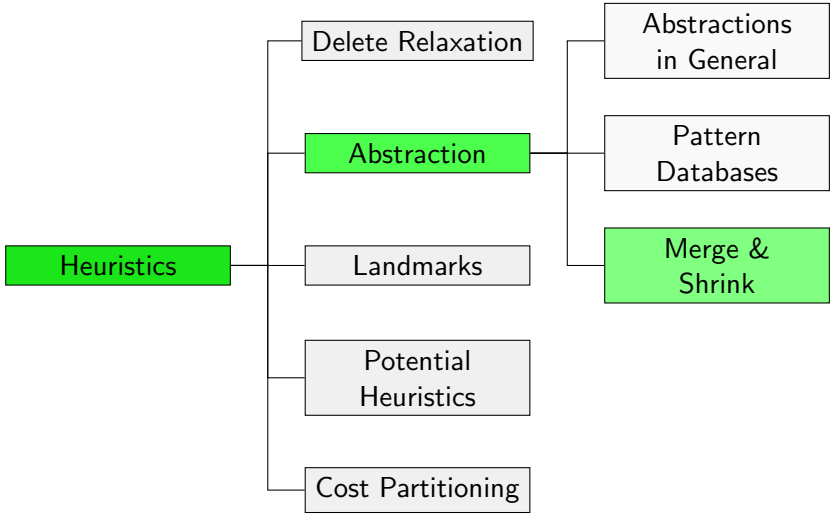
Universität Basel

November 7, 2018

Content of this Course

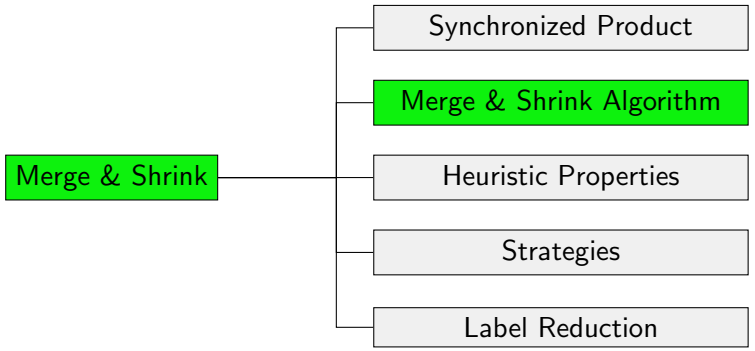


Content of this Course: Heuristics



D7.1 Generic Algorithm

Content of this Course: Merge & Shrink



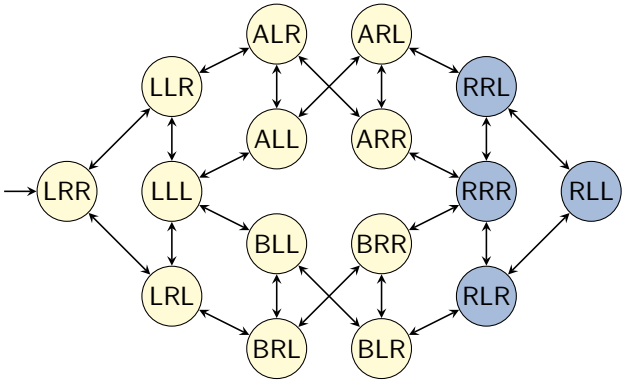
Generic Merge-and-shrink Abstractions: Outline

Using the results from the previous chapter, we can develop the ideas of a generic abstraction computation procedure that takes all state variables into account:

- Initialization step: Compute all abstract transition systems for atomic projections to form the initial abstraction collection.
- Merge steps: Combine two abstract systems in the collection by replacing them with their synchronized product. (Stop once only one transition system is left.)
- Shrink steps: If the abstractions in the collection are too large to compute their synchronized product, make them smaller by abstracting them further (applying an arbitrary abstraction to them).

We explain these steps with our running example.

Back to the Running Example

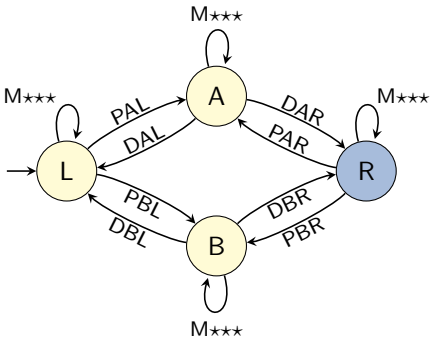


Logistics problem with one package, two trucks, two locations:

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

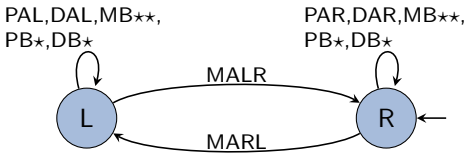
Initialization Step: Atomic Projection for Package

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



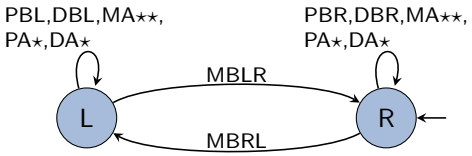
Initialization Step: Atomic Projection for Truck A

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



Initialization Step: Atomic Projection for Truck B

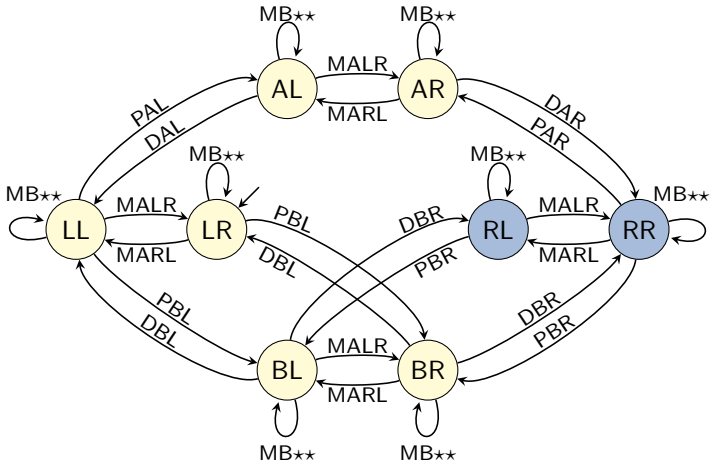
$\mathcal{T}^{\pi\{\text{truck B}\}}$:



current collection: $\{\mathcal{T}^{\pi\{\text{package}\}}, \mathcal{T}^{\pi\{\text{truck A}\}}, \mathcal{T}^{\pi\{\text{truck B}\}}\}$

First Merge Step

$\mathcal{T}_1 := \mathcal{T}^{\pi\{\text{package}\}} \otimes \mathcal{T}^{\pi\{\text{truck A}\}}$:



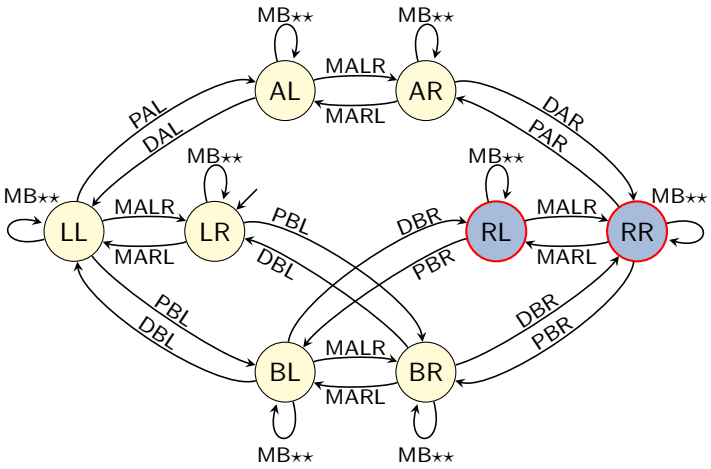
current collection: $\{\mathcal{T}_1, \mathcal{T}^{\pi\{\text{truck B}\}}\}$

Need to Simplify?

- If we have sufficient memory available, we can now compute $\mathcal{T}_1 \otimes \mathcal{T}^{\pi\{\text{truck B}\}}$, which would recover the complete transition system of the task.
- However, to illustrate the general idea, let us assume that we do not have sufficient memory for this product.
- More specifically, we will assume that after each product operation we need to reduce the result transition system to four states to obey memory constraints.
- So we need to reduce $\mathcal{T}_1$ to four states. We have a lot of leeway in deciding how exactly to abstract $\mathcal{T}_1$.
- In this example, we simply use an abstraction that leads to a good result in the end.

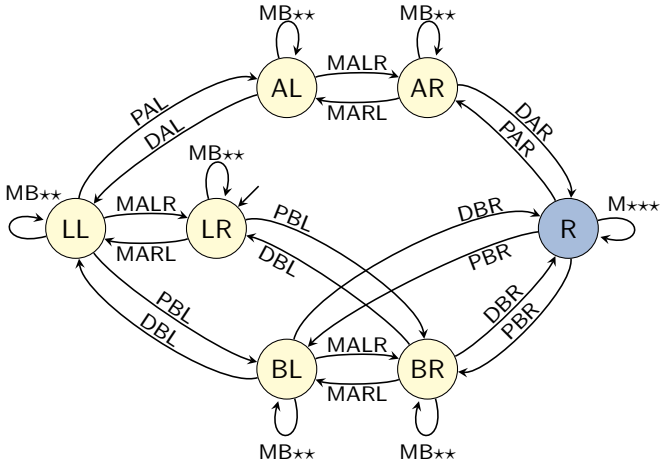
First Shrink Step

$\mathcal{T}_2 :=$ some abstraction of $\mathcal{T}_1$



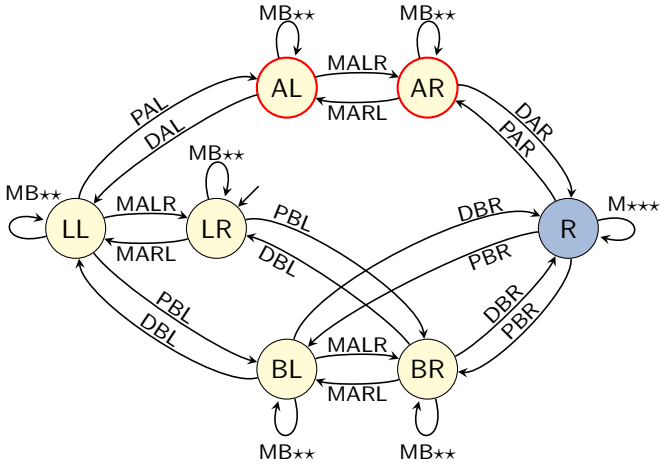
First Shrink Step

$\mathcal{T}_2 :=$ some abstraction of $\mathcal{T}_1$



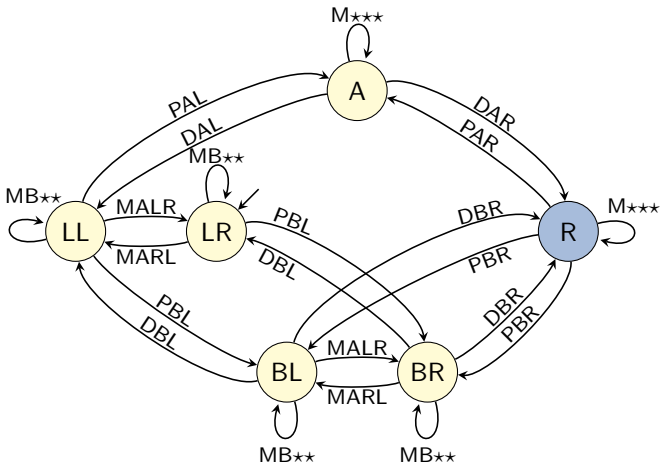
First Shrink Step

$\mathcal{T}_2 :=$ some abstraction of $\mathcal{T}_1$



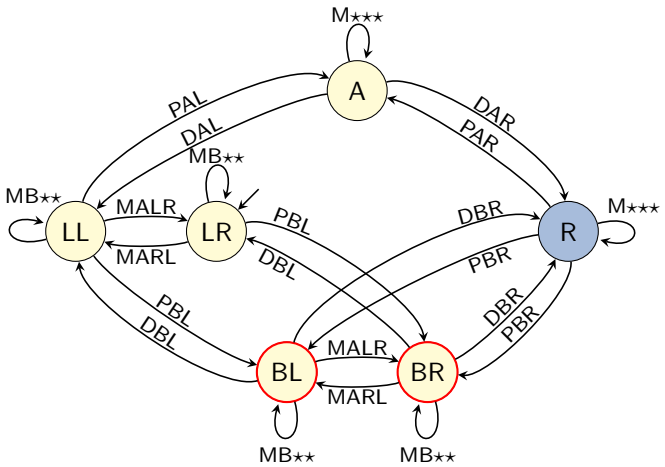
First Shrink Step

$\mathcal{T}_2 :=$ some abstraction of $\mathcal{T}_1$



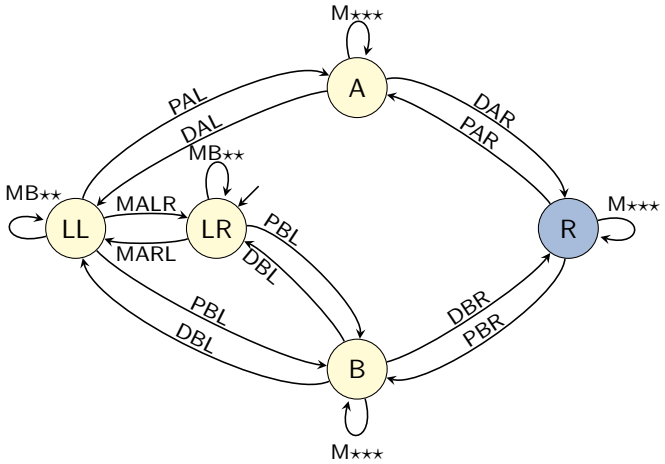
First Shrink Step

$\mathcal{T}_2 :=$ some abstraction of $\mathcal{T}_1$



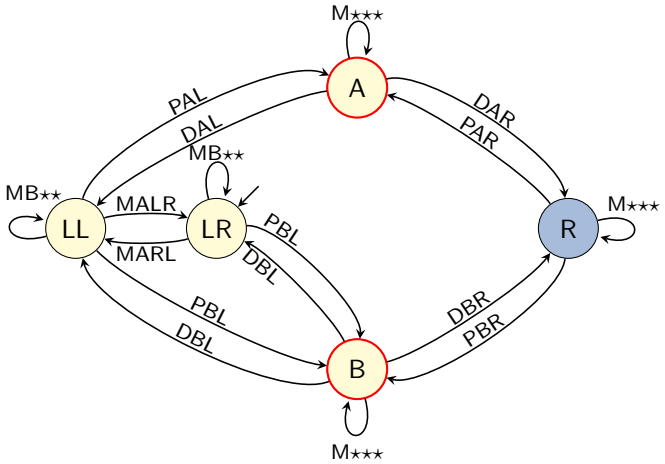
First Shrink Step

$\mathcal{T}_2 :=$ some abstraction of $\mathcal{T}_1$



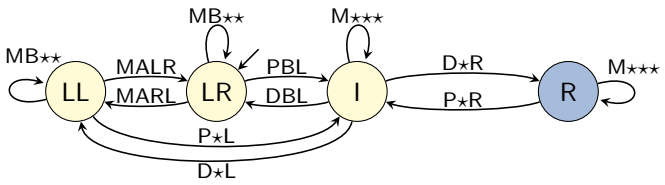
First Shrink Step

$\mathcal{T}_2 :=$ some abstraction of $\mathcal{T}_1$



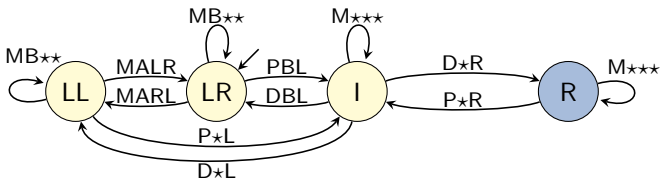
First Shrink Step

$\mathcal{T}_2 :=$ some abstraction of $\mathcal{T}_1$



First Shrink Step

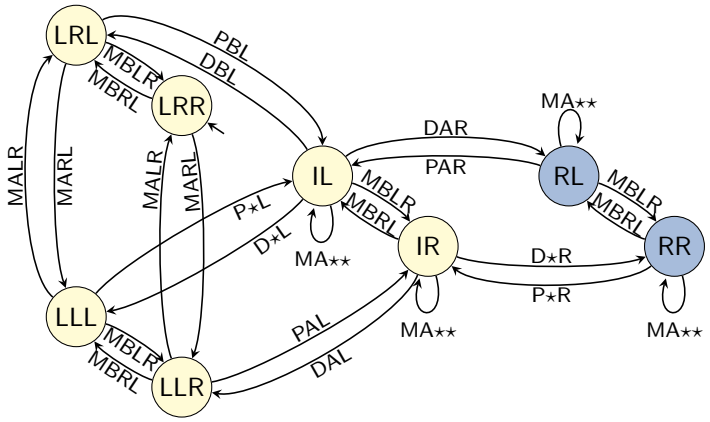
$\mathcal{T}_2 :=$ some abstraction of $\mathcal{T}_1$



current collection: $\{\mathcal{T}_2, \mathcal{T}^{\pi_{\{\text{truck B}\}}}\}$

Second Merge Step

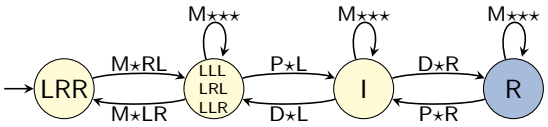
$\mathcal{T}_3 := \mathcal{T}_2 \otimes \mathcal{T}^{\pi_{\{\text{truck B}\}}}$:



current collection: $\{\mathcal{T}_3\}$

Another Shrink Step?

- Normally we could stop now and use the distances in the final abstract transition system as our heuristic function.
- However, if there were further state variables to integrate, we would simplify further, e.g. leading to the following abstraction (again with four states):



- We get a heuristic value of 3 for the initial state, better than any PDB heuristic that is a proper abstraction.
- The example generalizes to more locations and trucks, even if we stick to the size limit of 4 (after merging).

Generic Algorithm Template

Generic Merge & Shrink 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

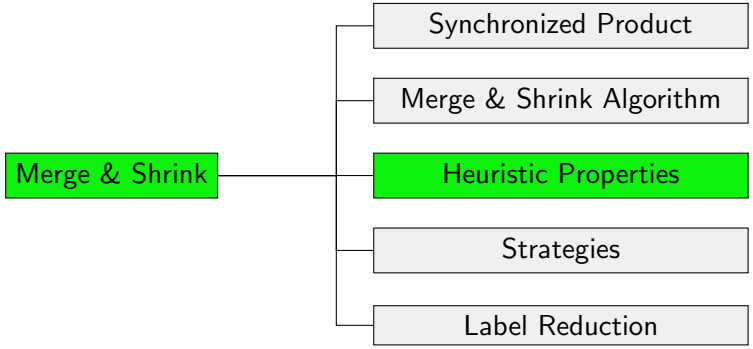
N: parameter bounding number of abstract states

Questions for practical implementation:

- Which abstractions to select? $\rightsquigarrow$ merging strategy
- How to shrink an abstraction? $\rightsquigarrow$ shrinking strategy
- How to choose N? $\rightsquigarrow$ usually: as high as memory allows

D7.2 Heuristic Properties

Content of this Course: Merge & Shrink



Heuristic Properties

- Each iteration of the algorithm corresponds to a transformation of the collection abs of transition systems.
- The exact transformation depends on the specific instantiation of the generic algorithm (e.g. of the merging and the shrinking strategy).
- For analyzing the properties of the resulting heuristic, we analyze properties of the individual transformations.

Collections of Transition Systems

Definition (Collection of Transition Systems)
A set X of transition systems is a collection of transition systems if all $\mathcal{T} \in X$ have the same set of labels and the same cost function. The combined system is $\mathcal{T}_X := \bigotimes_{\mathcal{T} \in X} \mathcal{T}$.

Remark: Strictly speaking, the combined system is not well-defined as the Cartesian product is neither commutative nor associative.

For our purpose it is sufficient that the results of all possible combination orders are isomorphic.

Safe Transformations

Definition (Safe Transformation)
Let X and X' be collections of transition systems with label sets L and L' and cost functions c and c', respectively.

The transformation from X to X' is safe if there exist functions σ and λ mapping the states and labels of $\mathcal{T}_X$ to the states and labels of $\mathcal{T}_{X'}$ such that

- $c'(\lambda(\ell)) \leq c(\ell)$ for all $\ell \in L$,
- if $\langle s, \ell, t \rangle$ is a transition of $\mathcal{T}_X$ then $\langle \sigma(s), \lambda(\ell), \sigma(t) \rangle$ is a transition of $\mathcal{T}_{X'}$, and
- if s is a goal state of $\mathcal{T}_X$ then $\sigma(s)$ is a goal state of $\mathcal{T}_{X'}$.

Examples

X: Collection of transition systems

Replacement with Synchronized Product is Safe
Let $\mathcal{T}_1, \mathcal{T}_2 \in X$ with $\mathcal{T}_1 \neq \mathcal{T}_2$. The transformation from X to $X' := (X \setminus \{\mathcal{T}_1, \mathcal{T}_2\}) \cup \{\mathcal{T}_1 \otimes \mathcal{T}_2\}$ is safe with $\sigma = \text{id}$ and $\lambda = \text{id}$.

Abstraction is Safe
Let α be an abstraction for $\mathcal{T}_i \in X$. The transformation from X to $X' := (X \setminus \{\mathcal{T}_i\}) \cup \{\mathcal{T}_i^\alpha\}$ is safe with $\lambda = \text{id}$ and $\sigma(\langle s_1, \ldots, s_n \rangle) = \langle s_1, \ldots, s_{i-1}, \alpha(s_i), s_{i+1}, \ldots, s_n \rangle$.

(Proofs omitted.)

Heuristic Properties (1)

Theorem
Let X and X' be collections of transition systems. If the transformation from X to X' is safe with functions σ and λ then $h(s) = h^*_{\mathcal{T}_{X'}}(\sigma(s))$ is a safe, goal-aware, admissible, and consistent heuristic for $\mathcal{T}_X$.

Proof.
We prove goal-awareness and consistency, the other properties follow from these two.

Goal-awareness: For all goal states $s_\star$ of $\mathcal{T}_X$, state $\sigma(s_\star)$ is a goal state of $\mathcal{T}_{X'}$ and therefore $h(s_\star) = h^*_{\mathcal{T}_{X'}}(\sigma(s_\star)) = 0$. ...

Heuristic Properties (2)

Proof (continued).

Consistency: Let c and c' be the label cost functions of X and X', respectively. Consider state s of $\mathcal{T}_X$ and transition $\langle s, \ell, t \rangle$.
As $\mathcal{T}_{X'}$ has a transition $\langle \sigma(s), \lambda(\ell), \sigma(t) \rangle$, it holds that

$$\begin{aligned} h(s) &= h^*_{\mathcal{T}_{X'}}(\sigma(s)) \\ &\leq c'(\lambda(\ell)) + h^*_{\mathcal{T}_{X'}}(\sigma(t)) \\ &= c'(\lambda(\ell)) + h(t) \\ &\leq c(\ell) + h(t) \end{aligned}$$

The second inequality holds due to the requirement on the label costs. □

Exact Transformations

Definition (Exact Transformation)

Let X and X' be collections of transition systems with label sets L and L' and cost functions c and c', respectively.

The transformation from X to X' is exact if there exist functions σ and λ mapping the states and labels of $\mathcal{T}_X$ to the states and labels of $\mathcal{T}_{X'}$ such that

1. σ and λ satisfy the requirements of safe transformations,
2. if $\langle s', \ell', t' \rangle$ is a transition of $\mathcal{T}_{X'}$ then $\langle s, \ell, t \rangle$ is a transition of $\mathcal{T}_X$ for all $s \in \sigma^{-1}(s')$, $t \in \sigma^{-1}(t')$ and some $\ell \in \lambda^{-1}(\ell')$,
3. if s' is a goal state of $\mathcal{T}_{X'}$ then all states $s \in \sigma^{-1}(s')$ are goal states of $\mathcal{T}_X$, and
4. $c(\ell) = c'(\lambda(\ell))$ for all $\ell \in L$.

$\rightsquigarrow$ no "new" transitions and goal states, no cheaper labels

Examples

Replacement with Synchronized Product is Exact

Let $\mathcal{T}_1, \mathcal{T}_2 \in X$ with $\mathcal{T}_1 \neq \mathcal{T}_2$. The transformation from X to $X' := (X \setminus \{\mathcal{T}_1, \mathcal{T}_2\}) \cup \{\mathcal{T}_1 \otimes \mathcal{T}_2\}$ is exact with $\sigma = \text{id}$ and $\lambda = \text{id}$.

(Proof omitted.)

More examples will follow.

Heuristic Properties with Exact Transformations (1)

Theorem

Let X and X' be collections of transition systems. If the transformation from X to X' is exact with functions σ and λ then $h^*_{\mathcal{T}_X}(s) = h^*_{\mathcal{T}_{X'}}(\sigma(s))$.

Proof.

As the transformation is safe, $h^*_{\mathcal{T}_{X'}}(\sigma(s))$ is admissible for $\mathcal{T}_X$ and therefore $h^*_{\mathcal{T}_X}(s) \geq h^*_{\mathcal{T}_{X'}}(\sigma(s))$.
For the other direction, we show that for every state s' of $\mathcal{T}_{X'}$ and goal path π' for s', there is for each $s \in \sigma^{-1}(s')$ a goal path in $\mathcal{T}_X$ that has the same cost. ...

Heuristic Properties with Exact Transformations (2)

Proof (continued).

Proof via induction over the length of π'.

$|\pi'| = 0$: If s' is a goal state of $\mathcal{T}_{X'}$ then each $s \in \sigma^{-1}(s')$ is a goal state of $\mathcal{T}_X$ and the empty path is a goal path for s in $\mathcal{T}_X$.

$|\pi'| = i + 1$: Let $\pi' = \langle s', \ell', t' \rangle \pi'_{t'}$, where $\pi'_{t'}$ is a goal path of length i from t'. Then there is for each $t \in \sigma^{-1}(t')$ a goal path π_t of the same cost in $\mathcal{T}_X$. Furthermore, for all $s \in \sigma^{-1}(s')$ there is a label $\ell \in \lambda^{-1}(\ell')$ such that $\mathcal{T}_X$ has a transition $\langle s, \ell, t \rangle$ with $t \in \sigma^{-1}(t')$. The path $\pi = \langle s, \ell, t \rangle \pi_t$ is a solution for s in $\mathcal{T}$. As ℓ and ℓ' must have the same cost and π_t and $\pi'_{t'}$ have the same cost, π has the same cost as π'. □

Sequences of Transformations

Theorem (Sequences of Transformations)

Let $X_1, \ldots, X_n$ be collections of transition systems.
If for $i \in \{1, \ldots, n-1\}$ the transformation from X_i to X_{i+1} is safe (exact) then the transformation from X_1 to X_n is safe (exact).

Proof idea: The composition of the σ and λ functions of each step satisfy the required conditions.

Consequences

Generic Merge & Shrink Algorithm

$abs := \{\mathcal{T}^{\pi\{v\}} \mid v \in V\} =: X_0$
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*

- Initially $\mathcal{T}_{abs}$ is the concrete transition system.
- Each iteration performs a safe transformation of *abs*.
  → the corresponding abstraction heuristic is safe, goal-aware, consistent, and admissible.
- If shrinking is exact, the corresponding heuristic is perfect.

D7.3 Summary

Summary

- Projections perfectly reflect a few state variables. Merge-and-shrink abstractions are a generalization that can reflect all state variables, but in a potentially lossy way.
- The merge steps combine two abstract transition systems by replacing them with their synchronized product.
- The shrink steps make an abstract system smaller by abstracting it further.
- As we only use safe transformations, the resulting heuristic is always admissible.
- If we use only exact transformations, the resulting heuristic is perfect.