

Planning and Optimization

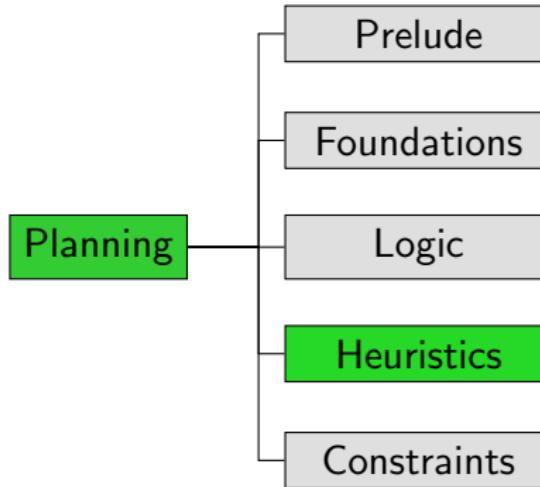
E11. Merge-and-Shrink: Properties and Shrinking Strategies

Malte Helmert and Gabriele Röger

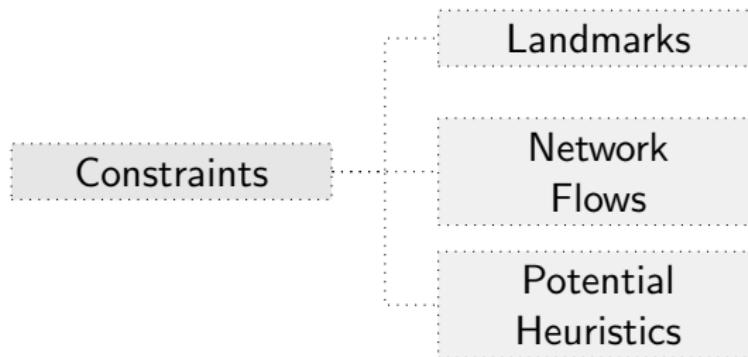
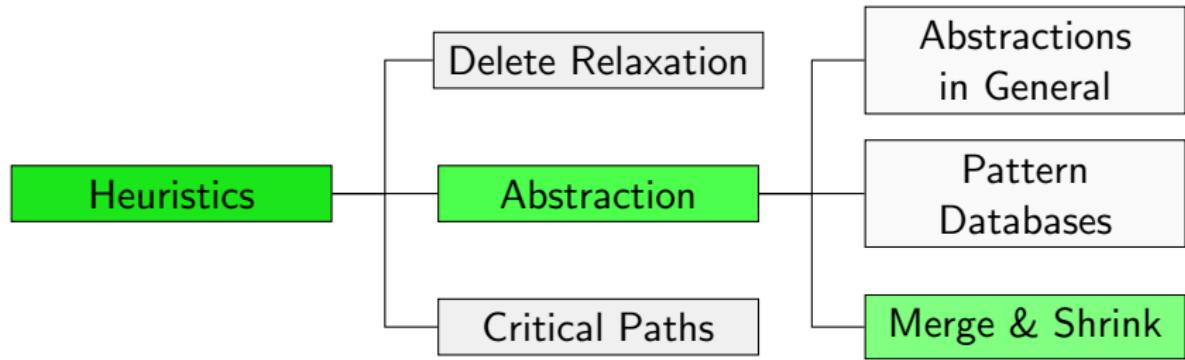
Universität Basel

November 21, 2022

Content of this Course



Content of this Course: Heuristics



Reminder: Generic Algorithm Template

Generic Merge & Shrink Algorithm for planning task Π

$F := F(\Pi)$

while $|F| > 1$:

select $type \in \{\text{merge, shrink}\}$

if $type = \text{merge}$:

select $\mathcal{T}_1, \mathcal{T}_2 \in F$

$F := (F \setminus \{\mathcal{T}_1, \mathcal{T}_2\}) \cup \{\mathcal{T}_1 \otimes \mathcal{T}_2\}$

if $type = \text{shrink}$:

select $\mathcal{T} \in F$

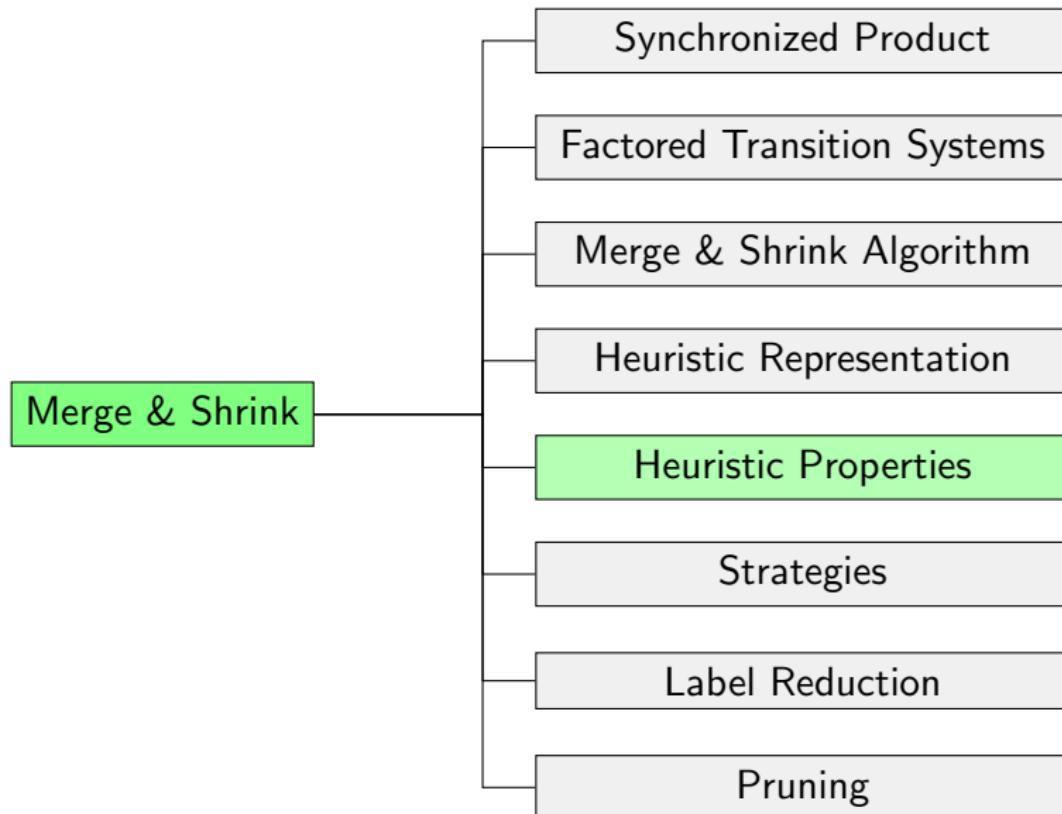
choose an abstraction mapping β on \mathcal{T}

$F := (F \setminus \{\mathcal{T}\}) \cup \{\mathcal{T}^\beta\}$

return the remaining factor \mathcal{T}^α in F

Heuristic Properties

Merge-and-Shrink



Properties of Merge-and-Shrink Heuristics

To understand merge-and-shrink abstractions better, we are interested in the **properties** of the resulting heuristic:

- Is it **admissible** ($h^\alpha(s) \leq h^*(s)$ for all states s)?
- Is it **consistent** ($h^\alpha(s) \leq c(o) + h^\alpha(t)$ for all trans. $s \xrightarrow{o} t$)?
- Is it **perfect** ($h^\alpha(s) = h^*(s)$ for all states s)?

Because merge-and-shrink is a **generic** procedure, the answers may depend on how exactly we instantiate it:

- size limits
- merge strategy
- shrink strategy

Merge-and-Shrink as Sequence of Transformations

- Consider a run of the merge-and-shrink construction algorithm with n iterations of the main loop.
- Let F_i ($0 \leq i \leq n$) be the FTS F after i loop iterations.
- Let \mathcal{T}_i ($0 \leq i \leq n$) be the transition system **represented** by F_i , i.e., $\mathcal{T}_i = \bigotimes F_i$.
- In particular, $F_0 = F(\Pi)$ and $F_n = \{\mathcal{T}_n\}$.
- For SAS⁺ tasks Π , we also know $\mathcal{T}_0 = \mathcal{T}(\Pi)$.

For a formal study, it is useful to view merge-and-shrink construction as a sequence of **transformations** from \mathcal{T}_i to \mathcal{T}_{i+1} .

(We do it in a bit more general fashion than necessary for merge and shrink steps only, to also cover some improvements we will see later.)

Transformations

Definition (Transformation)

Let $\mathcal{T} = \langle S, L, c, T, s_0, S_* \rangle$ and $\mathcal{T}' = \langle S', L', c', T', s'_0, S'_* \rangle$ be transition systems.

Let $\sigma : S \rightarrow S'$ map the states of \mathcal{T} to the states of \mathcal{T}' and $\lambda : L \rightarrow L'$ map the labels of \mathcal{T} to the labels of \mathcal{T}' .

The tuple $\tau = \langle \mathcal{T}, \sigma, \lambda, \mathcal{T}' \rangle$ is called a **transformation** from \mathcal{T} to \mathcal{T}' . We also write it as $\mathcal{T} \xrightarrow{\sigma, \lambda} \mathcal{T}'$.

The transformation τ induces the **heuristic h^τ** for \mathcal{T} defined as $h^\tau(s) = h_{\mathcal{T}'}^*(\sigma(s))$.

Example: If α is an abstraction mapping for transition system \mathcal{T} , then $\mathcal{T} \xrightarrow{\alpha, \text{id}} \mathcal{T}^\alpha$ is a transformation.

Conservative Transformations

Definition (Conservative Transformation)

Let \mathcal{T} and \mathcal{T}' be transition systems with label sets L and L' and cost functions c and c' , respectively.

A transformation $\langle \mathcal{T}, \sigma, \lambda, \mathcal{T}' \rangle$ is **conservative** if

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

Example: If α is an abstraction mapping for transition system \mathcal{T} , then $\mathcal{T} \xrightarrow{\alpha, \text{id}} \mathcal{T}^\alpha$ is a conservative transformation.

Conservative Transformations: Heuristic Properties (1)

Theorem

*If τ is a **conservative transformation** from transition system \mathcal{T} to transition system \mathcal{T}' then h^τ is a **safe, consistent, goal-aware and admissible** heuristic for \mathcal{T} .*

Proof.

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

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

Conservative Transformations: Heuristic Properties (2)

Proof (continued).

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

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

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

Exact Transformations

Definition (Exact Transformation)

Let \mathcal{T} and \mathcal{T}' be transition systems with label sets L and L' and cost functions c and c' , respectively.

A transformation $\langle \mathcal{T}, \sigma, \lambda, \mathcal{T}' \rangle$ is **exact** if it is conservative and

- ① if $\langle s', \ell', t' \rangle$ is a transition of \mathcal{T}' then for all $s \in \sigma^{-1}(s')$ there is a transition $\langle s, \ell, t \rangle$ of \mathcal{T} with $t \in \sigma^{-1}(t')$ and $\ell \in \lambda^{-1}(\ell')$,
- ② if s' is a goal state of \mathcal{T}' then all states $s \in \sigma^{-1}(s')$ are goal states of \mathcal{T} , and
- ③ $c(\ell) = c'(\lambda(\ell))$ for all $\ell \in L$.

⇒ no “new” transitions and goal states, no cheaper labels

Heuristic Properties with Exact Transformations (1)

Theorem

If τ is an *exact transformation* from transition system \mathcal{T} to transition system \mathcal{T}' then h^τ is the perfect heuristic h^* for \mathcal{T} .

Proof.

As the transformation is conservative, h^τ is admissible for \mathcal{T}' and therefore $h_{\mathcal{T}'}^*(s) \geq h^\tau(s)$.

For the other direction, we show that for every state s' of \mathcal{T}' and goal path π' for s' , there is for each $s \in \sigma^{-1}(s')$ a goal path in \mathcal{T} 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}' then each $s \in \sigma^{-1}(s')$ is a goal state of \mathcal{T} and the empty path is a goal path for s in \mathcal{T} .

$|\pi'| = i + 1$: 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} (by ind. hypothesis). Furthermore, for all $s \in \sigma^{-1}(s')$ there is a state $t \in \sigma^{-1}(t')$ and a label $\ell \in \lambda^{-1}(\ell')$ such that \mathcal{T} has a transition $\langle s, \ell, t \rangle$. 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' . □

Composing Transformations

Merge-and-shrink performs many transformations in sequence.
We can formalize this with a notion of **composition**:

- Given $\tau = \mathcal{T} \xrightarrow{\sigma, \lambda} \mathcal{T}'$ and $\tau' = \mathcal{T}' \xrightarrow{\sigma', \lambda'} \mathcal{T}''$,
their **composition** $\tau'' = \tau' \circ \tau$ is defined as
 $\tau'' = \mathcal{T} \xrightarrow{\sigma' \circ \sigma, \lambda' \circ \lambda} \mathcal{T}''$.
- If τ and τ' are conservative, then $\tau' \circ \tau$ is conservative.
- If τ and τ' are exact, then $\tau' \circ \tau$ is exact.

Merge-and-Shrink Transformations

F : factored transition system

Replacement with Synchronized Product is Conservative and Exact

Let $\mathcal{T}_1, \mathcal{T}_2 \in F$ with $\mathcal{T}_1 \neq \mathcal{T}_2$.

Let $F' := (X \setminus \{\mathcal{T}_1, \mathcal{T}_2\}) \cup \{\mathcal{T}_1 \otimes \mathcal{T}_2\}$.

Then there is an exact transformation $\langle \otimes F, \sigma, \text{id}, \otimes F' \rangle$.

Up to the isomorphism we know from the synchronized product, we can use $\sigma = \text{id}$.

Abstraction is Conservative

Let α be an abstraction of $\mathcal{T}_i \in F$ and let $F' := (F \setminus \{\mathcal{T}_i\}) \cup \{\mathcal{T}_i^\alpha\}$.

The transformation $\langle \otimes F, \sigma, \text{id}, \otimes F' \rangle$ with

$\sigma(\langle s_1, \dots, s_n \rangle) = \langle s_1, \dots, s_{i-1}, \alpha(s_i), s_{i+1}, \dots, s_n \rangle$ is conservative.

(Proofs omitted.)

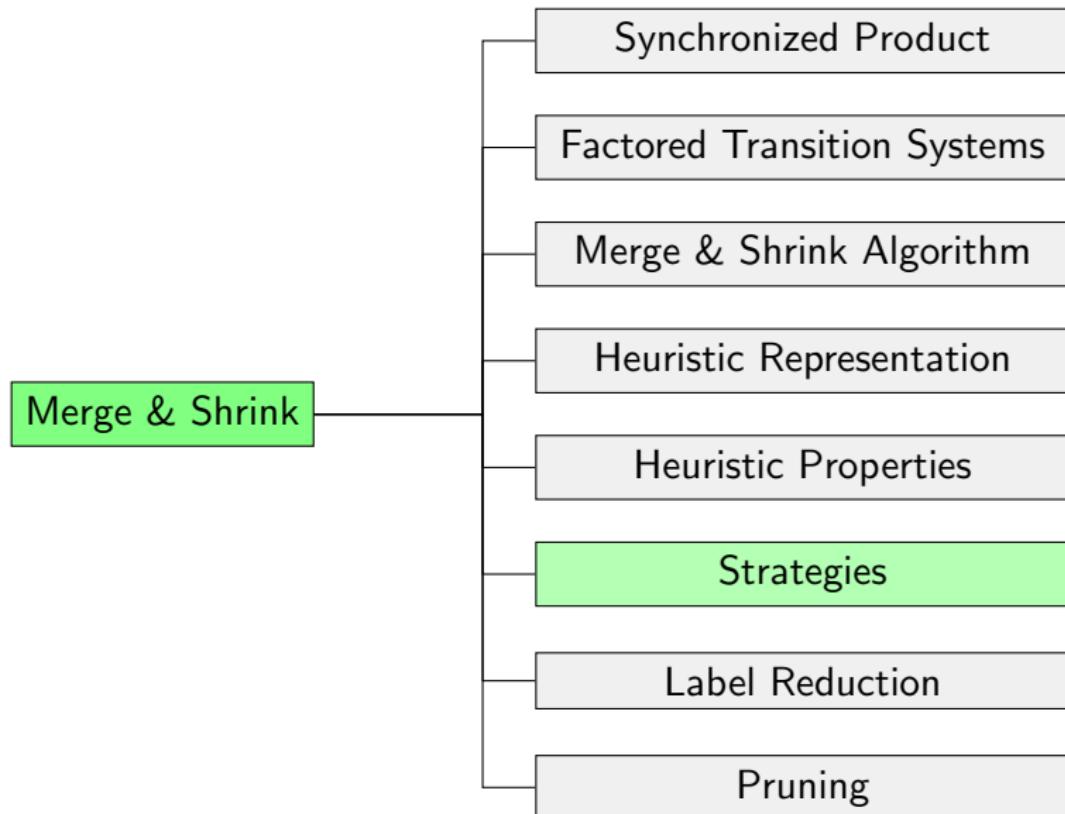
Properties of Merge-and-Shrink Heuristics

We can conclude the following properties of merge-and-shrink heuristics for SAS⁺ tasks:

- The heuristic is always **admissible** and **consistent** (because it is induced by a composition of conservative transformations).
- If all shrink transformation used are exact, the heuristic is **perfect** (because it is induced by a composition of exact transformations).

Shrinking Strategies

Merge-and-Shrink



Reminder: Generic Algorithm Template

```
 $F := F(\Pi)$ 
while  $|F| > 1$ :
  select  $type \in \{\text{merge, shrink}\}$ 
  if  $type = \text{merge}$ :
    select  $\mathcal{T}_1, \mathcal{T}_2 \in F$ 
     $F := (F \setminus \{\mathcal{T}_1, \mathcal{T}_2\}) \cup \{\mathcal{T}_1 \otimes \mathcal{T}_2\}$ 
  if  $type = \text{shrink}$ :
    select  $\mathcal{T} \in F$ 
    choose an abstraction mapping  $\beta$  on  $\mathcal{T}$ 
     $F := (F \setminus \{\mathcal{T}\}) \cup \{\mathcal{T}^\beta\}$ 
return the remaining factor  $\mathcal{T}^\alpha$  in  $F$ 
```

Remaining Questions:

- Which abstractions to select for merging? \rightsquigarrow merging strategy
- How to shrink an abstraction? \rightsquigarrow shrinking strategy

Shrinking Strategies

How to shrink an abstraction?

We cover two common approaches:

- f -preserving shrinking
- bisimulation-based shrinking

f -preserving Shrinking Strategy

f -preserving Shrinking Strategy

Repeatedly combine abstract states with
identical abstract goal distances (h values) and
identical abstract initial state distances (g values).

Rationale: preserves heuristic value and overall graph shape

Tie-breaking Criterion

Prefer combining states where $g + h$ is high.

In case of ties, combine states where h is high.

Rationale: states with high $g + h$ values are less likely to be explored by A^* , so inaccuracies there matter less

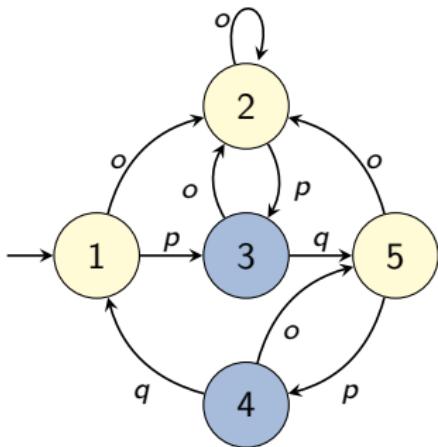
Bisimulation

Definition (Bisimulation)

Let $\mathcal{T} = \langle S, L, c, T, s_0, S_* \rangle$ be a transition system. An equivalence relation \sim on S is a **bisimulation** for \mathcal{T} if for every $\langle s, \ell, s' \rangle \in T$ and every $t \sim s$ there is a transition $\langle t, \ell, t' \rangle \in T$ with $t' \sim s'$.

A bisimulation \sim is **goal-respecting** if $s \sim t$ implies that either $s, t \in S_*$ or $s, t \notin S_*$.

Bisimulation: Example



\sim with equivalence classes $\{\{1, 2, 5\}, \{3, 4\}\}$ is a goal-respecting bisimulation.

Bisimulation Abstractions

Definition (Abstractions as Bisimulation)

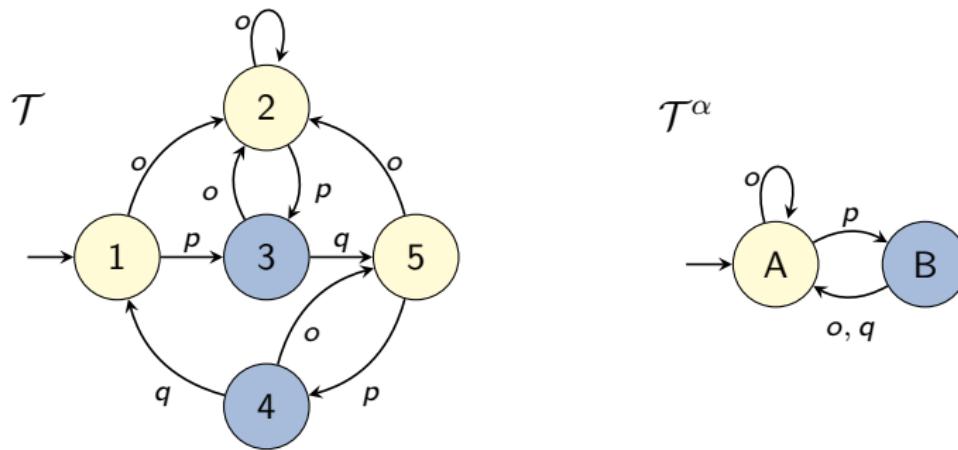
Let $\mathcal{T} = \langle S, L, c, T, s_0, S_* \rangle$ be a transition system and $\alpha : S \rightarrow S'$ be an abstraction of \mathcal{T} . The abstraction induces the equivalence relation \sim_α as $s \sim_\alpha t$ iff $\alpha(s) = \alpha(t)$.

We say that α is a (goal-respecting) bisimulation for \mathcal{T} if \sim_α is a (goal-respecting) bisimulation for \mathcal{T} .

Abstraction as Bisimulations: Example

Abstraction α with

$\alpha(1) = \alpha(2) = \alpha(5) = A$ and $\alpha(3) = \alpha(4) = B$
is a goal-respecting bisimulation for \mathcal{T} .



Goal-respecting Bisimulations are Exact

Theorem

Let F be a factored transition system and α be an abstraction of $\mathcal{T}_i \in F$.

If α is a goal-respecting bisimulation then the transformation $\langle \otimes F, \sigma, id, \otimes F' \rangle$ with

- $\sigma(\langle s_1, \dots, s_n \rangle) = \langle s_1, \dots, s_{i-1}, \alpha(s_i), s_{i+1}, \dots, s_n \rangle$ and
- $F' := (F \setminus \{\mathcal{T}_i\}) \cup \{\mathcal{T}_i^\alpha\}$

is exact.

(Proofs omitted.)

Shrinking with bisimulation preserves the heuristic estimates.

Bisimulations: Discussion

- As all bisimulations preserve all relevant information, we are interested in the **coarsest** such abstraction (to shrink as much as possible).
- There is always a unique coarsest bisimulation for \mathcal{T} and it can be computed efficiently (from the explicit representation).
- In some cases, computing the bisimulation is still too expensive or it cannot sufficiently shrink a transition system.

Summary

Summary

- Merge-and-shrink abstractions can be analyzed by viewing them as a sequence of **transformations**.
- We only use **conservative transformations**, and hence merge-and-shrink heuristics for SAS^+ tasks are **admissible** and **consistent**.
- Merge-and-shrink heuristics for SAS^+ tasks that only use **exact** transformations are **perfect**.
- **Bisimulation** is an **exact** shrinking method.