Planning and Optimization

D1. Abstractions: Formal Definition and Heuristics

Gabriele Röger and Thomas Keller

Universität Basel

October 29, 2018

D1.1 Abstractions

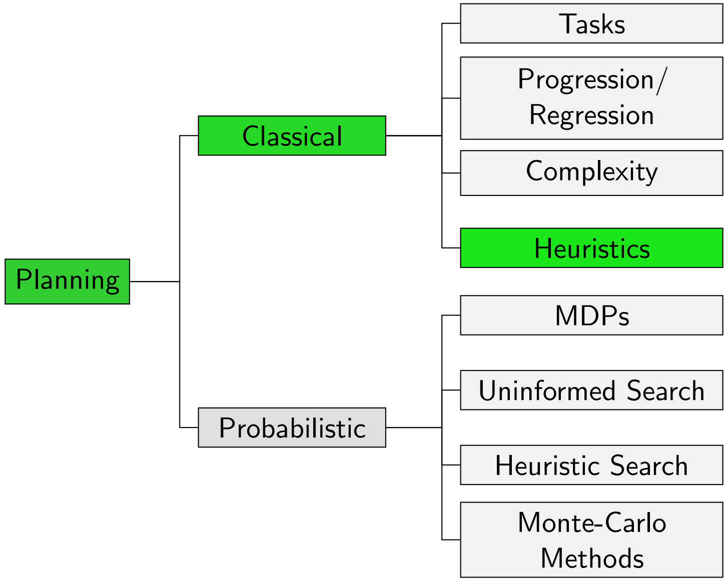
D1.2 Homomorphisms and Isomorphisms

D1.3 Abstraction Heuristics

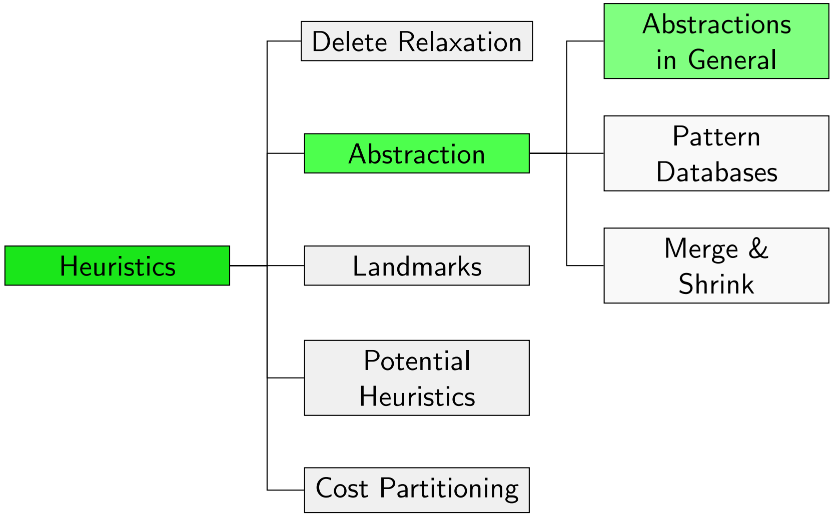
D1.4 Coarsenings and Refinements

D1.5 Summary

Content of this Course



Content of this Course: Heuristics



D1.1 Abstractions

Abstracting a Transition System

Abstracting a transition system means dropping some distinctions between states, while preserving the transition behaviour as much as possible.

- An abstraction of a transition system $\mathcal{T}$ is defined by an abstraction mapping α that defines which states of $\mathcal{T}$ should be distinguished and which ones should not.
- From $\mathcal{T}$ and α, we compute an abstract transition system $\mathcal{T}^\alpha$ which is similar to $\mathcal{T}$, but smaller.
- The abstract goal distances (goal distances in $\mathcal{T}^\alpha$) are used as heuristic estimates for goal distances in $\mathcal{T}$.

Computing the Abstract Transition System

Given $\mathcal{T}$ and α, how do we compute $\mathcal{T}^\alpha$?

Requirement

We want to obtain an admissible heuristic.
Hence, $h^*(\alpha(s))$ (in the abstract state space $\mathcal{T}^\alpha$) should never overestimate $h^*(s)$ (in the concrete state space $\mathcal{T}$).

An easy way to achieve this is to ensure that all solutions in $\mathcal{T}$ are also present in $\mathcal{T}^\alpha$:

- If s is a goal state in $\mathcal{T}$, then $\alpha(s)$ is a goal state in $\mathcal{T}^\alpha$.
- If $\mathcal{T}$ has a transition from s to t, then $\mathcal{T}^\alpha$ has a transition from $\alpha(s)$ to $\alpha(t)$.

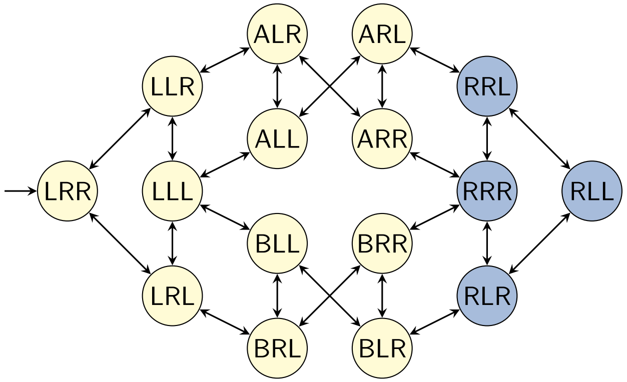
Example Task: One Package, Two Trucks

Example (One Package, Two Trucks)

Consider the following FDR planning task $\langle V, I, O, \gamma\rangle$:

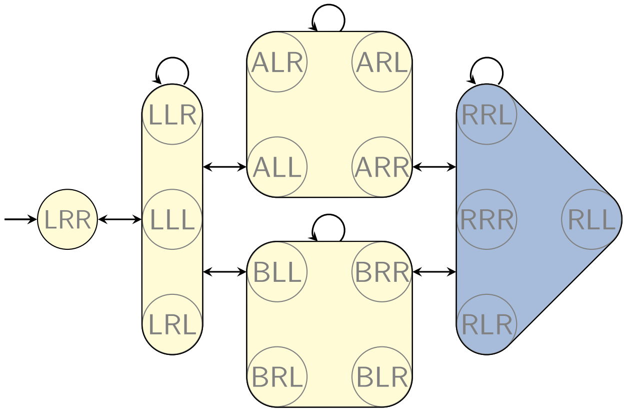
- $V = \{p, t_\mathsf{A}, t_\mathsf{B}\}$ with
  - $\mathrm{dom}(p) = \{\mathsf{L}, \mathsf{R}, \mathsf{A}, \mathsf{B}\}$
  - $\mathrm{dom}(t_\mathsf{A}) = \mathrm{dom}(t_\mathsf{B}) = \{\mathsf{L}, \mathsf{R}\}$
- $I = \{p \mapsto \mathsf{L}, t_\mathsf{A} \mapsto \mathsf{R}, t_\mathsf{B} \mapsto \mathsf{R}\}$
- $O = \{\mathrm{pickup}_{i,j} \mid i \in \{\mathsf{A}, \mathsf{B}\}, j \in \{\mathsf{L}, \mathsf{R}\}\}$
  $\cup \{\mathrm{drop}_{i,j} \mid i \in \{\mathsf{A}, \mathsf{B}\}, j \in \{\mathsf{L}, \mathsf{R}\}\}$
  $\cup \{\mathrm{move}_{i,j,j'} \mid i \in \{\mathsf{A}, \mathsf{B}\}, j, j' \in \{\mathsf{L}, \mathsf{R}\}, j \neq j'\}$, where
  - $\mathrm{pickup}_{i,j} = \langle t_i = j \land p = j, p := i, 1\rangle$
  - $\mathrm{drop}_{i,j} = \langle t_i = j \land p = i, p := j, 1\rangle$
  - $\mathrm{move}_{i,j,j'} = \langle t_i = j, t_i := j', 1\rangle$
- $\gamma = (p = \mathsf{R})$

Concrete Transition System of Example Task



- State $\{p \mapsto i, t_\mathsf{A} \mapsto j, t_\mathsf{B} \mapsto k\}$ is depicted as ijk.
- Transition labels are again not shown. For example, the transition from LLL to ALL has the label pickup$_{\mathsf{A},\mathsf{L}}$.

Abstract Transition System of Example Task



- State $\{p \mapsto i, t_\mathsf{A} \mapsto j, t_\mathsf{B} \mapsto k\}$ is depicted as ijk.
- Transition labels are again not shown. For example, the transition from LLL to ALL has the label pickup$_{\mathsf{A},\mathsf{L}}$.

Abstractions

Definition (Abstraction)

Let $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$ be a transition system.

An abstraction (also: abstraction function, abstraction mapping) of $\mathcal{T}$ is a function $\alpha : S \to S^\alpha$ defined on the states of $\mathcal{T}$, where S^α is an arbitrary set.

Without loss of generality, we require that α is surjective.

Intuition: α maps the states of $\mathcal{T}$ to another (usually smaller) abstract state space.

Abstract Transition System

Definition (Abstract Transition System)

Let $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$ be a transition system, and let $\alpha : S \to S^\alpha$ be an abstraction of $\mathcal{T}$.

The abstract transition system induced by α, in symbols $\mathcal{T}^\alpha$, is the transition system $\mathcal{T}^\alpha = \langle S^\alpha, L, c, T^\alpha, s_0^\alpha, S_\star^\alpha \rangle$ defined by:

- $T^\alpha = \{\langle \alpha(s), \ell, \alpha(t) \rangle \mid \langle s, \ell, t \rangle \in T\}$
- $s_0^\alpha = \alpha(s_0)$
- $S_\star^\alpha = \{\alpha(s) \mid s \in S_\star\}$

Terminology

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

- $\mathcal{T}$ is called the concrete transition system.
- $\mathcal{T}^\alpha$ is called the abstract transition system.
- Similarly: concrete/abstract state space, concrete/abstract transition, etc.

Practical Requirements for Abstractions

To be useful in practice, an abstraction heuristic must be efficiently computable. This gives us two requirements for α:

- For a given state s, the abstract state $\alpha(s)$ must be efficiently computable.
- For a given abstract state $\alpha(s)$, the abstract goal distance $h^*(\alpha(s))$ must be efficiently computable.

There are a number of ways of achieving these requirements:

- pattern database heuristics (Culberson & Schaeffer, 1996)
- merge-and-shrink abstractions (Dräger, Finkbeiner & Podelski, 2006)
- Cartesian abstractions (Ball, Podelski & Rajamani, 2001)
- structural patterns (Katz & Domshlak, 2008b)

D1.2 Homomorphisms and Isomorphisms

Homomorphisms and Isomorphisms

- The abstraction mapping α that transforms $\mathcal{T}$ to $\mathcal{T}^\alpha$ is also called a strict homomorphism from $\mathcal{T}$ to $\mathcal{T}^\alpha$.
- Roughly speaking, in mathematics a homomorphism is a property-preserving mapping between structures.
- A strict homomorphism is one where no additional features are introduced. A non-strict homomorphism in planning would mean that the abstract transition system may include additional transitions and goal states not induced by α.
- We only consider strict homomorphisms in this course.
- If α is bijective, it is called an isomorphism between $\mathcal{T}$ and $\mathcal{T}^\alpha$, and the two transition systems are called isomorphic.

Isomorphic Transition Systems

The notion of isomorphic transition systems is important enough to warrant a formal definition:

Definition (Isomorphic Transition Systems)

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

We say that $\mathcal{T}$ is isomorphic to $\mathcal{T}'$, in symbols $\mathcal{T} \sim \mathcal{T}'$, if there exist bijective functions $\varphi : S \to S'$ and $\lambda : L \to L'$ such that:

- $s \xrightarrow{\ell} t \in T$ iff $\varphi(s) \xrightarrow{\lambda(\ell)} \varphi(t) \in T'$,
- $c'(\lambda(\ell)) = c(\ell)$ for all $\ell \in L$,
- $\varphi(s_0) = s'_0$, and
- $s \in S_\star$ iff $\varphi(s) \in S'_\star$.

Graph-Equivalent Transition Systems

Sometimes a weaker notion of equivalence is useful:

Definition (Graph-Equivalent Transition Systems)

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

We say that $\mathcal{T}$ is graph-equivalent to $\mathcal{T}'$, in symbols $\mathcal{T} \overset{\mathrm{G}}{\sim} \mathcal{T}'$, if there exists a bijective function $\varphi : S \to S'$ such that:

- There is a transition $s \xrightarrow{\ell} t \in T$ with $c(\ell) = k$ iff there is a transition $\varphi(s) \xrightarrow{\ell'} \varphi(t) \in T'$ with $c'(\ell') = k$,
- $\varphi(s_0) = s'_0$, and
- $s \in S_\star$ iff $\varphi(s) \in S'_\star$.

Note: The labels of $\mathcal{T}$ and $\mathcal{T}'$ do not matter except that transitions of the same cost must be preserved.

Isomorphism vs. Graph Equivalence

- $(\sim)$ and $(\overset{\mathrm{G}}{\sim})$ are equivalence relations.
- Two isomorphic transition systems are interchangeable for all practical intents and purposes.
- Two graph-equivalent transition systems are interchangeable for most intents and purposes.
- In particular, their goal distances are identical.
- Isomorphism implies graph equivalence, but not vice versa.

D1.3 Abstraction Heuristics

Abstraction Heuristics

Definition (Abstraction Heuristic)

Let $\alpha : S \to S^\alpha$ be an abstraction of a transition system $\mathcal{T}$.

The abstraction heuristic induced by α, written h^α, is the heuristic function $h^\alpha : S \to \mathbb{R}_0^+ \cup \{\infty\}$ defined as

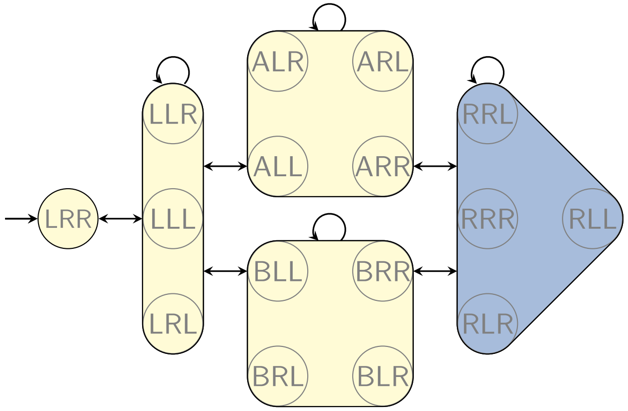
$$h^\alpha(s) = h^*_{\mathcal{T}^\alpha}(\alpha(s)) \quad \text{for all } s \in S,$$

where $h^*_{\mathcal{T}^\alpha}$ denotes the goal distance function in $\mathcal{T}^\alpha$.

Notes:

- $h^\alpha(s) = \infty$ if no goal state of $\mathcal{T}^\alpha$ is reachable from $\alpha(s)$
- We also apply abstraction terminology to planning tasks Π, which stand for their induced transition systems. For example, an abstraction of Π is an abstraction of $\mathcal{T}(\Pi)$.

Abstraction Heuristics: Example



$h^\alpha(\{p \mapsto \mathsf{L}, t_\mathsf{A} \mapsto \mathsf{R}, t_\mathsf{B} \mapsto \mathsf{R}\}) = 3$

Consistency of Abstraction Heuristics (1)

Theorem (Consistency and Admissibility of h^α)

Let α be an abstraction of a transition system $\mathcal{T}$.
Then h^α is safe, goal-aware, admissible and consistent.

Proof.

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

Let $\mathcal{T} = \langle S, L, c, T, s_0, S_\star \rangle$.
Let $\mathcal{T}^\alpha = \langle S^\alpha, L, c, T^\alpha, s_0^\alpha, S_\star^\alpha \rangle$.

Goal-awareness: We need to show that $h^\alpha(s) = 0$ for all $s \in S_\star$, so let $s \in S_\star$. Then $\alpha(s) \in S_\star^\alpha$ by the definition of abstract transition systems, and hence $h^\alpha(s) = h^*_{\mathcal{T}^\alpha}(\alpha(s)) = 0$. ...

Consistency of Abstraction Heuristics (2)

Proof (continued).

Consistency: Consider any state transition $s \xrightarrow{\ell} t$ of $\mathcal{T}$.
We need to show $h^\alpha(s) \leq c(\ell) + h^\alpha(t)$.

By the definition of $\mathcal{T}^\alpha$, we get $\alpha(s) \xrightarrow{\ell} \alpha(t) \in \mathcal{T}^\alpha$.
Hence, $\alpha(t)$ is a successor of $\alpha(s)$ in $\mathcal{T}^\alpha$ via the label ℓ.

We get:

$$\begin{aligned} h^\alpha(s) &= h^*_{\mathcal{T}^\alpha}(\alpha(s)) \\ &\leq c(\ell) + h^*_{\mathcal{T}^\alpha}(\alpha(t)) \\ &= c(\ell) + h^\alpha(t), \end{aligned}$$

where the inequality holds because perfect goal distances $h^*_{\mathcal{T}^\alpha}$ are consistent in $\mathcal{T}^\alpha$.
(The shortest path from $\alpha(s)$ to the goal in $\mathcal{T}^\alpha$ cannot be longer than the shortest path from $\alpha(s)$ to the goal via $\alpha(t)$.) □

D1.4 Coarsenings and Refinements

Abstractions of Abstractions

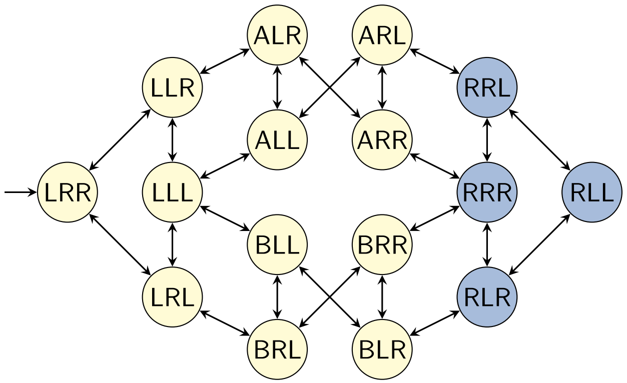
Since abstractions map transition systems to transition systems, they are composable:

- Using a first abstraction $\alpha : S \to S'$, map $\mathcal{T}$ to $\mathcal{T}^{\alpha}$.
- Using a second abstraction $\beta : S' \to S''$, map $\mathcal{T}^{\alpha}$ to $(\mathcal{T}^{\alpha})^{\beta}$.

The result is the same as directly using the abstraction $(\beta \circ \alpha)$:

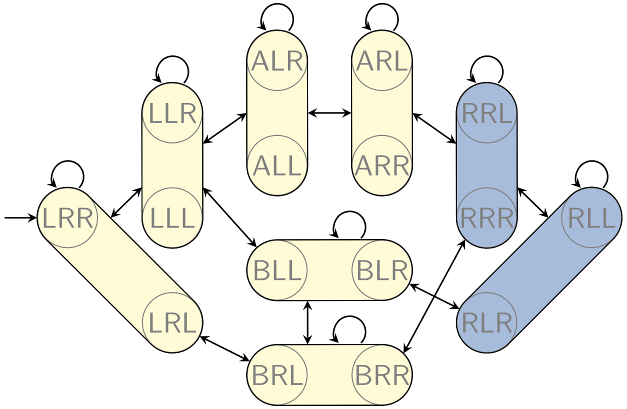
- Let $\gamma : S \to S''$ be defined as $\gamma(s) = (\beta \circ \alpha)(s) = \beta(\alpha(s))$.
- Then $\mathcal{T}^{\gamma} = (\mathcal{T}^{\alpha})^{\beta}$.

Abstractions of Abstractions: Example (1)



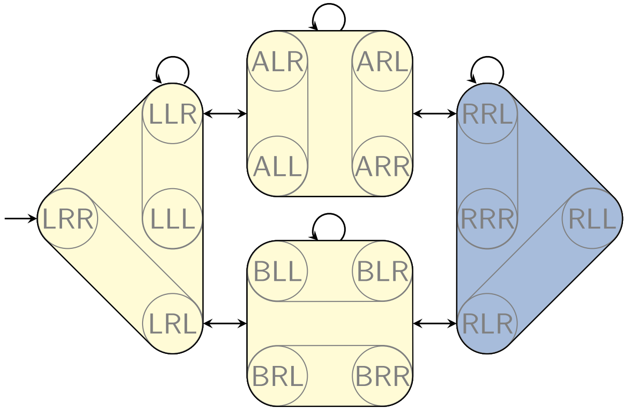
transition system $\mathcal{T}$

Abstractions of Abstractions: Example (2)



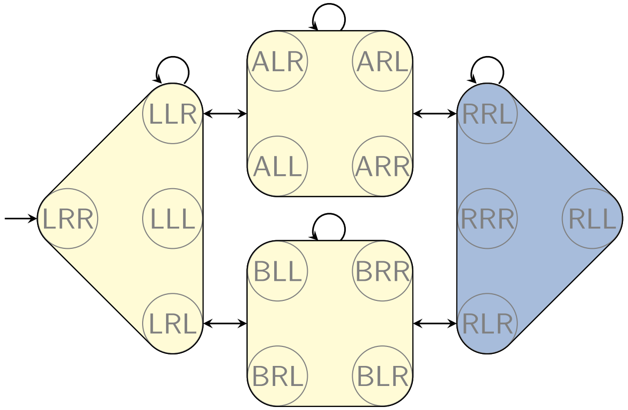
Transition system $\mathcal{T}'$ as an abstraction of $\mathcal{T}$ (ignore t_{B})

Abstractions of Abstractions: Example (3)



Transition system $\mathcal{T}''$ as an abstraction of $\mathcal{T}'$
(ignore t_A)

Abstractions of Abstractions: Example (3)



Transition system $\mathcal{T}''$ as an abstraction of $\mathcal{T}$
(ignore t_A and t_B)

Coarsenings and Refinements

Definition (Coarsening and Refinement)

Let α and γ be abstractions of the same transition system such that $\gamma = \beta \circ \alpha$ for some function β.

Then γ is called a coarsening of α and α is called a refinement of γ.

Heuristic Quality of Refinements

Theorem (Heuristic Quality of Refinements)

Let α and γ be abstractions of the same transition system such that α is a refinement of γ.

Then h^α dominates h^γ.

In other words, $h^\gamma(s) \leq h^\alpha(s) \leq h^*(s)$ for all states s.

Heuristic Quality of Refinements: Proof

Proof.

Since α is a refinement of γ,
there exists a function β with $\gamma = \beta \circ \alpha$.

For all states s of Π, we get:

$$\begin{aligned} h^{\gamma}(s) &= h^*_{\mathcal{T}^{\gamma}}(\gamma(s)) \\ &= h^*_{\mathcal{T}^{\gamma}}(\beta(\alpha(s))) \\ &= h^{\beta}_{\mathcal{T}^{\alpha}}(\alpha(s)) \\ &\leq h^*_{\mathcal{T}^{\alpha}}(\alpha(s)) \\ &= h^{\alpha}(s), \end{aligned}$$

where the inequality holds because $h^{\beta}_{\mathcal{T}^{\alpha}}$ is an admissible heuristic in the transition system $\mathcal{T}^{\alpha}$. □

D1.5 Summary

Summary

- Abstraction is one of the principled ways of deriving heuristics.
- An abstraction is a function α that maps the states S of a transition system to another (usually smaller) set S^{α}.
- This induces an abstract transition system $\mathcal{T}^{\alpha}$, which behaves like the original transition system $\mathcal{T}$ except that states mapped to the same abstract state cannot be distinguished.
- Abstractions α induce abstraction heuristics h^{α}: $h^{\alpha}(s)$ is the goal distance of $\alpha(s)$ in the abstract transition system.
- Abstraction heuristics are safe, goal-aware, admissible and consistent.
- Abstractions can be composed, leading to coarser vs. finer abstractions. Heuristics for finer abstractions dominate those for coarser ones.