

Planning and Optimization

E5. Abstractions: Additive Abstractions

Malte Helmert and Gabriele Röger

Universität Basel

November 10, 2025

Planning and Optimization

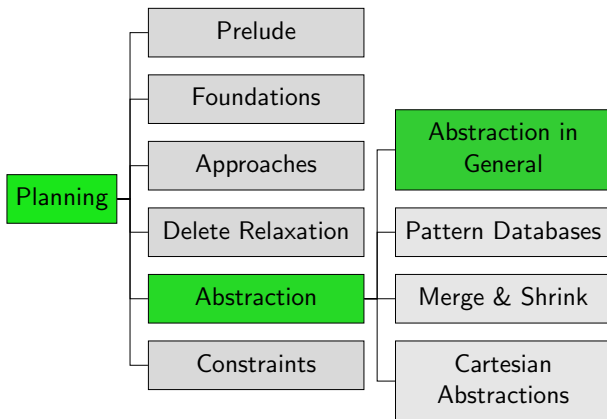
November 10, 2025 — E5. Abstractions: Additive Abstractions

E5.1 Additivity

E5.2 Outlook

E5.3 Summary

Content of the Course



E5.1 Additivity

Orthogonality of Abstractions

Definition (Orthogonal)

Let α_1 and α_2 be abstractions of transition system $\mathcal{T}$.

We say that α_1 and α_2 are **orthogonal** if for all transitions $s \xrightarrow{\ell} t$ of $\mathcal{T}$, we have $\alpha_1(s) = \alpha_1(t)$ or $\alpha_2(s) = \alpha_2(t)$.

Affecting Transition Labels

Definition (Affecting Transition Labels)

Let $\mathcal{T}$ be a transition system, and let ℓ be one of its labels.

We say that ℓ **affects** $\mathcal{T}$ if $\mathcal{T}$ has a transition $s \xrightarrow{\ell} t$ with $s \neq t$.

Theorem (Affecting Labels vs. Orthogonality)

Let α_1 and α_2 be abstractions of transition system $\mathcal{T}$.

*If no label of $\mathcal{T}$ affects both $\mathcal{T}^{\alpha_1}$ and $\mathcal{T}^{\alpha_2}$,
then α_1 and α_2 are orthogonal.*

(Easy proof omitted.)

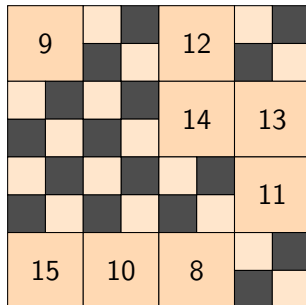
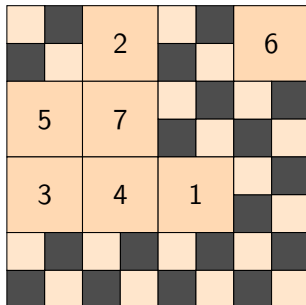
Orthogonal Abstractions: Example

	2		6
5	7		
3	4	1	

9		12	
		14	13
			11
15	10	8	

Are the abstractions orthogonal?

Orthogonal Abstractions: Example



Are the abstractions orthogonal?

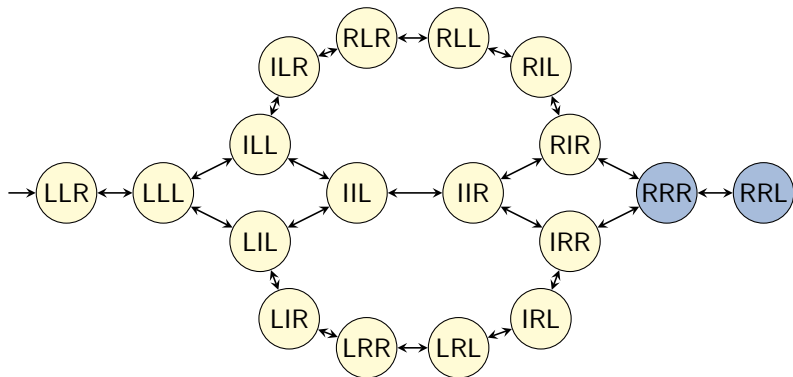
Orthogonality and Additivity

Theorem (Additivity for Orthogonal Abstractions)

Let $h^{\alpha_1}, \dots, h^{\alpha_n}$ be abstraction heuristics of the same transition system such that α_i and α_j are orthogonal for all $i \neq j$.

Then $\sum_{i=1}^n h^{\alpha_i}$ is a safe, goal-aware, admissible and consistent heuristic for Π .

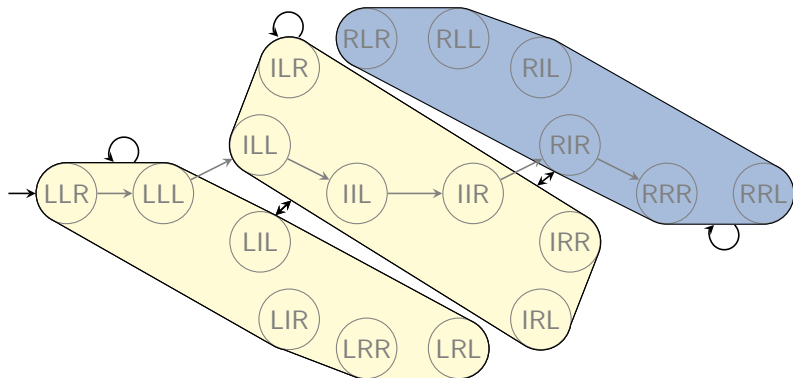
Orthogonality and Additivity: Example



transition system $\mathcal{T}$

state variables: first package, second package, truck

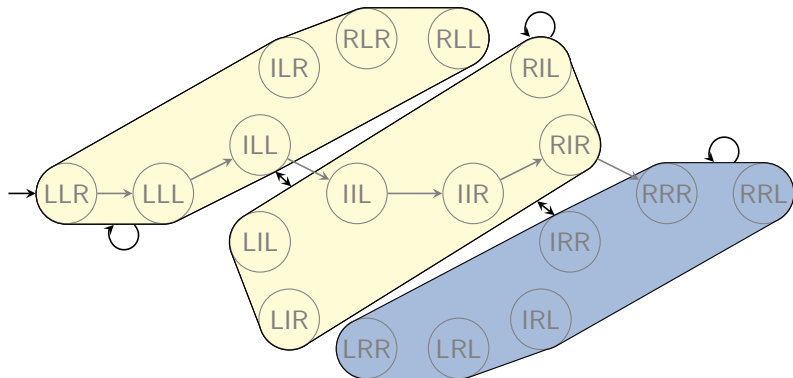
Orthogonality and Additivity: Example



abstraction α_1

abstraction: only consider value of first package

Orthogonality and Additivity: Example



abstraction α_2 (orthogonal to α_1)

abstraction: only consider value of second package

Orthogonality and Additivity: Proof (1)

Proof.

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

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

Let $h = \sum_{i=1}^n h^{\alpha_i}$.

Goal-awareness: For goal states $s \in S_\star$,

$h(s) = \sum_{i=1}^n h^{\alpha_i}(s) = \sum_{i=1}^n 0 = 0$ because all individual
abstraction heuristics are goal-aware.

...

Orthogonality and Additivity: Proof (2)

Proof (continued).

Consistency: Let $s \xrightarrow{o} t \in \mathcal{T}$. We must prove $h(s) \leq c(o) + h(t)$.

Because the abstractions are orthogonal, $\alpha_i(s) \neq \alpha_i(t)$
for **at most one** $i \in \{1, \dots, n\}$.

Case 1: $\alpha_i(s) = \alpha_i(t)$ for all $i \in \{1, \dots, n\}$.

$$\begin{aligned} \text{Then } h(s) &= \sum_{i=1}^n h^{\alpha_i}(s) \\ &= \sum_{i=1}^n h_{\mathcal{T}^{\alpha_i}}^*(\alpha_i(s)) \\ &= \sum_{i=1}^n h_{\mathcal{T}^{\alpha_i}}^*(\alpha_i(t)) \\ &= \sum_{i=1}^n h^{\alpha_i}(t) \\ &= h(t) \leq c(o) + h(t). \end{aligned}$$

...

Orthogonality and Additivity: Proof (3)

Proof (continued).

Case 2: $\alpha_i(s) \neq \alpha_i(t)$ for exactly one $i \in \{1, \dots, n\}$.

Let $k \in \{1, \dots, n\}$ such that $\alpha_k(s) \neq \alpha_k(t)$.

$$\begin{aligned} \text{Then } h(s) &= \sum_{i=1}^n h^{\alpha_i}(s) \\ &= \sum_{i \in \{1, \dots, n\} \setminus \{k\}} h_{\mathcal{T}^{\alpha_i}}^*(\alpha_i(s)) + h^{\alpha_k}(s) \\ &\leq \sum_{i \in \{1, \dots, n\} \setminus \{k\}} h_{\mathcal{T}^{\alpha_i}}^*(\alpha_i(t)) + c(o) + h^{\alpha_k}(t) \\ &= c(o) + \sum_{i=1}^n h^{\alpha_i}(t) \\ &= c(o) + h(t), \end{aligned}$$

where the inequality holds because $\alpha_i(s) = \alpha_i(t)$ for all $i \neq k$ and h^{α_k} is consistent. □

E5.2 Outlook

Using Abstraction Heuristics in Practice

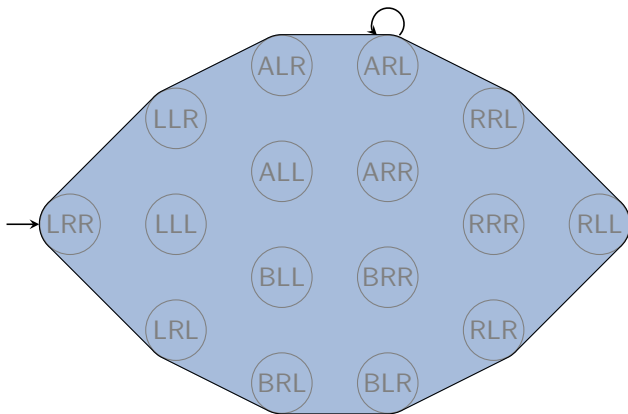
In practice, there are conflicting goals for abstractions:

- ▶ we want to obtain an **informative heuristic**, but
- ▶ want to keep its **representation small**.

Abstractions have small representations if

- ▶ there are **few abstract states** and
- ▶ there is a **succinct encoding for α** .

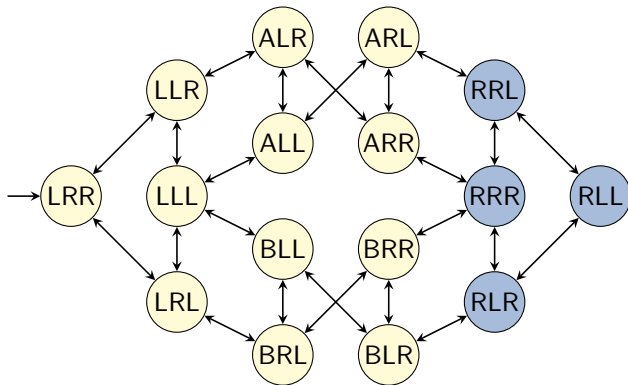
Counterexample: One-State Abstraction



One-state abstraction: $\alpha(s) := \text{const.}$

- + very few abstract states and succinct encoding for α
- completely uninformative heuristic

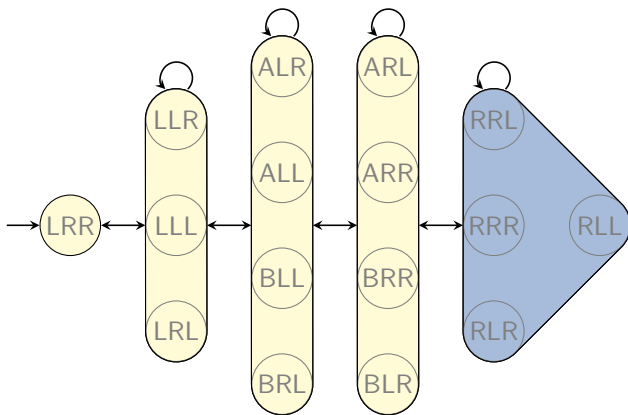
Counterexample: Identity Abstraction



Identity abstraction: $\alpha(s) := s$.

- + perfect heuristic and succinct encoding for α
- too many abstract states

Counterexample: Perfect Abstraction



Perfect abstraction: $\alpha(s) := h^*(s)$.

- + perfect heuristic and usually few abstract states
- usually no succinct encoding for α

Automatically Deriving Good Abstraction Heuristics

Abstraction Heuristics for Planning: Main Research Problem
Automatically derive effective abstraction heuristics
for planning tasks.

~> we will study three state-of-the-art approaches
in the following chapters

E5.3 Summary

Summary

- ▶ Abstraction heuristics from **orthogonal** abstractions can be **added** without losing admissibility or consistency.
- ▶ One sufficient condition for orthogonality is that all abstractions are **affected** by **disjoint** sets of **labels**.
- ▶ Practically useful abstractions are those which give **informative heuristics**, yet have a **small representation**.
- ▶ Coming up with **good abstractions automatically** is the main research challenge when applying abstraction heuristics in planning.