Automated Planning

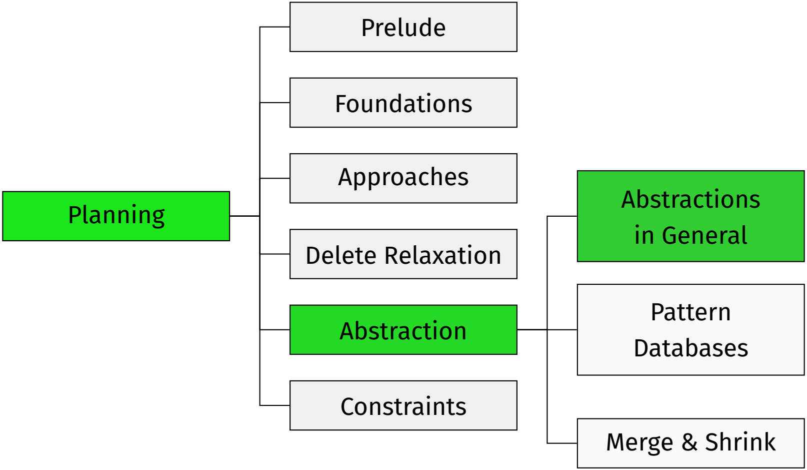
E5. Abstractions: Orthogonality and Additivity

Jendrik Seipp

Linköping University

based on slides from the AI group at the University of Basel

Content of this Course



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.

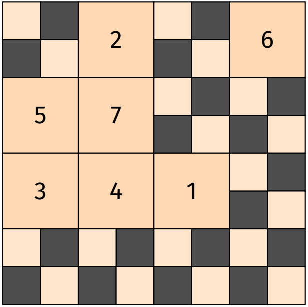
Orthogonal Abstractions: Example

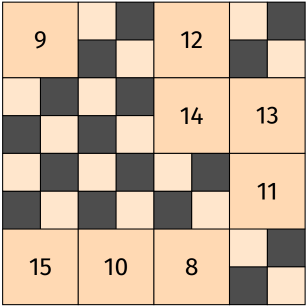
	2		6
5	7		
3	4	1	
			■

9		12	
		14	13
			11
15	10	8	■

Are the abstractions orthogonal? ⇝ No, because the blank is considered both in $\mathcal{T}^{\alpha_1}$ and $\mathcal{T}^{\alpha_2}$.

Orthogonal Abstractions: Example





Are the abstractions orthogonal? ⇝ Yes.

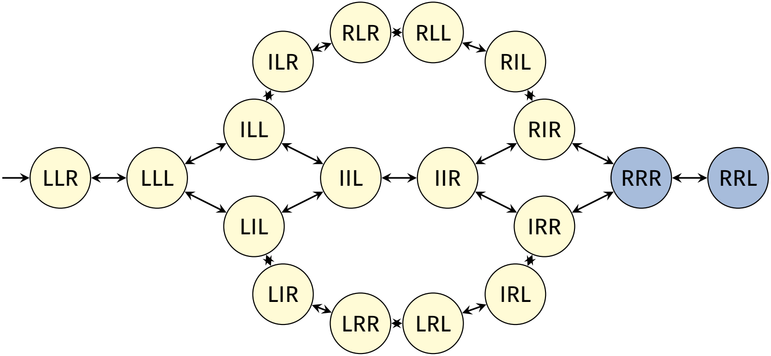
Orthogonality and Additivity

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.

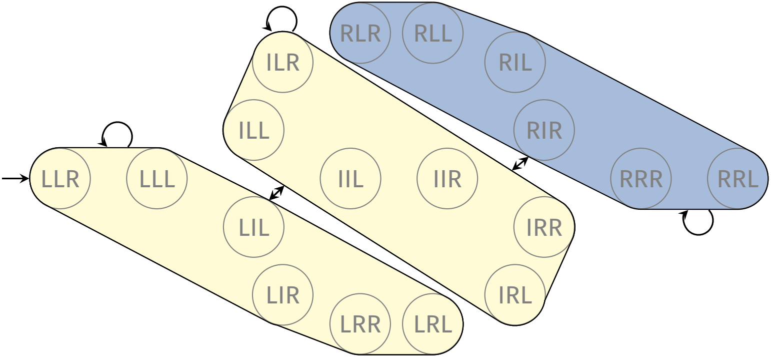
Orthogonality and Additivity: Example



transition system $\mathcal{T}$

state variables: first package, second package, truck

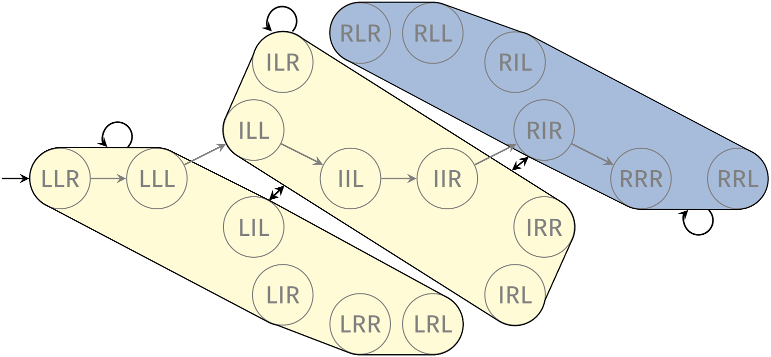
Orthogonality and Additivity: Example



abstraction α_1

abstraction: only consider value of first package

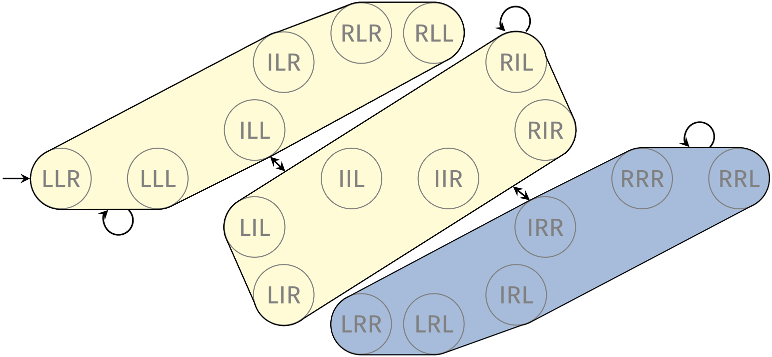
Orthogonality and Additivity: Example



abstraction α_1

abstraction: only consider value of first package

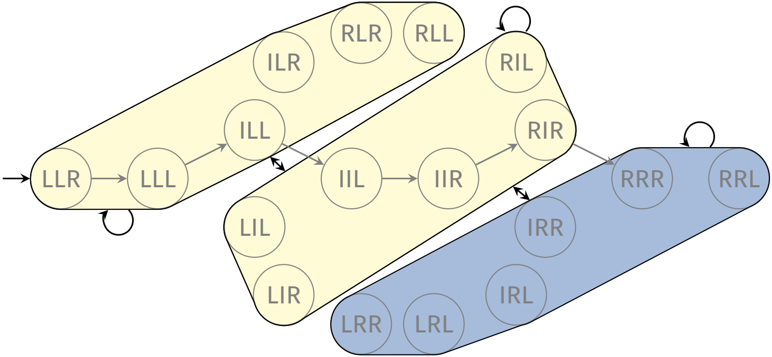
Orthogonality and Additivity: Example



abstraction α_2 (orthogonal to α_1)

abstraction: only consider value of second package

Orthogonality and Additivity: Example



abstraction α_2 (orthogonal to α_1)

abstraction: only consider value of second package

Outlook

Using Abstraction Heuristics in Practice

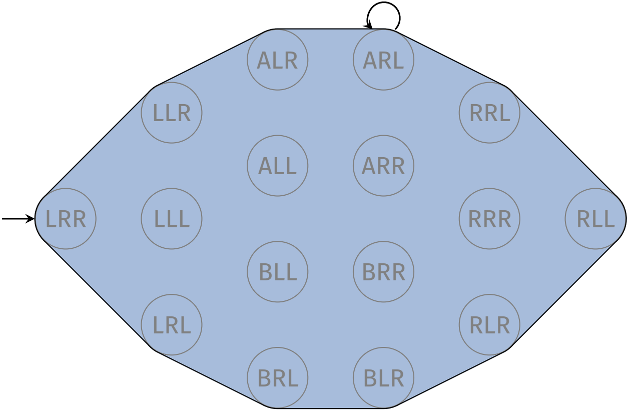
In practice, there are conflicting goals for abstractions:

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

Abstractions have small representations if

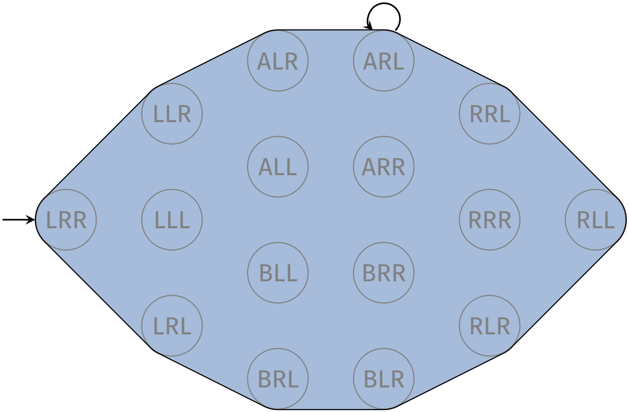
- there are few abstract states and
- there is a succinct encoding for α.

Counterexample: One-State Abstraction



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

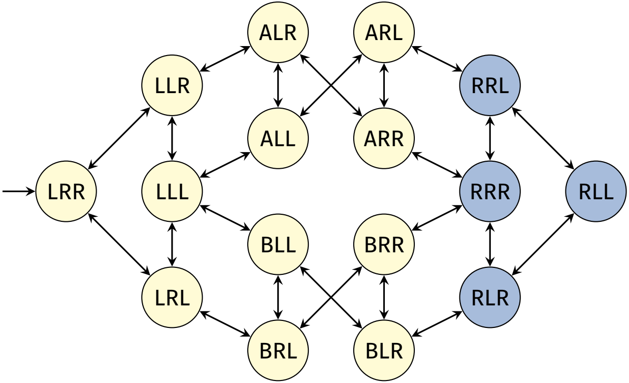
Counterexample: One-State Abstraction



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

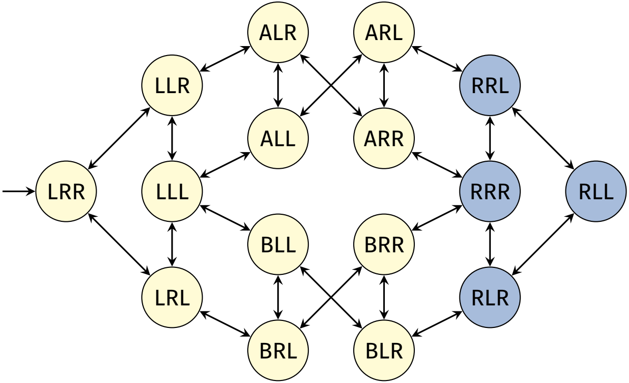
- + very few abstract states and succinct encoding for α
- − completely uninformative heuristic

Counterexample: Identity Abstraction



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

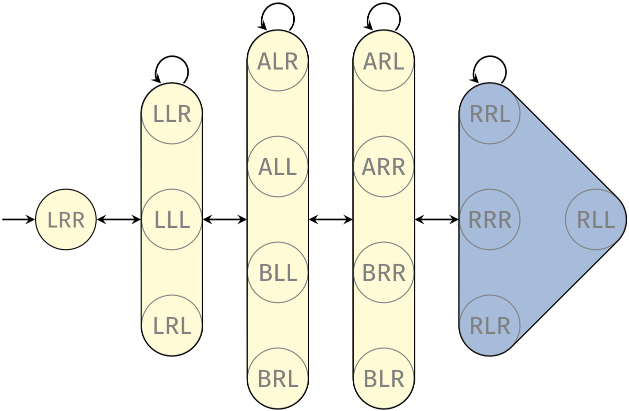
Counterexample: Identity Abstraction



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

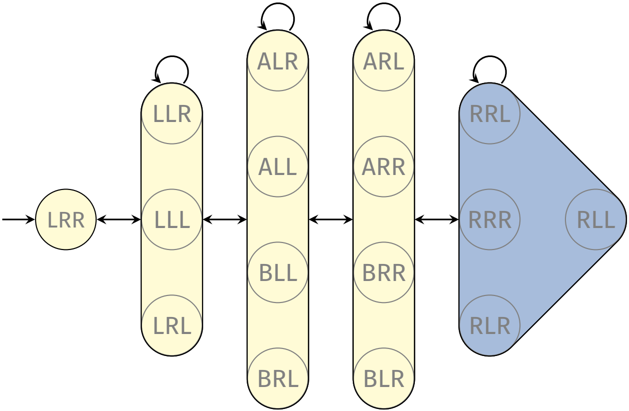
- + perfect heuristic and succinct encoding for α
- – too many abstract states

Counterexample: Perfect Abstraction



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

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 two state-of-the-art approaches
in the following chapters

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.