

Grundlagen der Künstlichen Intelligenz

34. Handlungsplanung: Abstraktion und Musterdatenbanken

Malte Helmert

Universität Basel

12. Mai 2014

Planungsheuristiken

Wir besprechen **drei Kernideen** für allgemeine Heuristiken:

- Delete-Relaxierung
- **Abstraktion** \rightsquigarrow heute
- Landmarken

Planungsheuristiken

Wir besprechen **drei Kernideen** für allgemeine Heuristiken:

- Delete-Relaxierung
- **Abstraktion** \rightsquigarrow heute
- Landmarken

Grundidee Abstraktion

Schätze Lösungskosten durch Betrachten
einer **kleineren** Planungsaufgabe ab.

Handlungsplanung: Überblick

Kapitelüberblick:

- 30. Einführung
- 31. Planungsformalismen
- 32.–33. Planungsheuristiken: Delete-Relaxierung
- 34.–35. Planungsheuristiken: Abstraktion
 - 34. **Abstraktion und Musterdatenbanken**
 - 35. Merge-and-Shrink-Abstraktionen
- 36.–37. Planungsheuristiken: Landmarken

SAS⁺
●○○○○○

Abstraktionen
○○○○○○○○○○

Musterdatenbanken
○○○○○○○○

Zusammenfassung
○○

SAS⁺

SAS⁺-Kodierung

- in diesem und nächstem Kapitel: SAS⁺-Kodierung statt STRIPS (siehe Kapitel 31)
- Unterschied: Zustandsvariablen nicht alle binär, sondern mit **endlichem Wertebereich** $\text{dom}(v)$
- entsprechend Vorbedingungen, Effekte und Ziele als **partielle Belegungen** gegeben
- sonst alles gleich wie STRIPS

(Praktische Planer konvertieren automatisch zwischen STRIPS und SAS⁺.)

SAS⁺-Planungsaufgabe

Definition (SAS⁺-Planungsaufgabe)

Eine SAS⁺-Planungsaufgabe ist ein 5-Tupel $\Pi = \langle V, \text{dom}, I, G, A \rangle$ mit folgenden Komponenten:

- V : endliche Menge von **Zustandsvariablen**
- dom : **Wertebereiche**; $\text{dom}(v)$ für $v \in V$ endlich, nicht-leer
 - Zustände sind **totale Belegungen** für V gemäss dom
- I : der **Anfangszustand** (Zustand = totale Belegung)
- G : **Ziele** (partielle Belegung)
- A : endliche Menge von **Aktionen**, jeweils mit:
 - $\text{pre}(a)$: ihre **Vorbedingungen** (partielle Belegung)
 - $\text{eff}(a)$ ihre **Effekte** (partielle Belegung)
 - $\text{cost}(a) \in \mathbb{N}_0$: ihre **Kosten**

Zustandsraum zu einer SAS⁺-Planungsaufgabe

Definition (von SAS⁺-Planungsaufgabe induz. Zustandsraum)

Sei $\Pi = \langle V, \text{dom}, I, G, A \rangle$ eine SAS⁺-Planungsaufgabe.

Dann **induziert** Π den **Zustandsraum** $\mathcal{S}(\Pi) = \langle S, A, \text{cost}, T, s_0, S_* \rangle$:

- **Zustandsmenge:** totale Belegungen von V gemäss dom
- **Aktionen:** die Aktionen A von Π
- **Aktionskosten:** cost ist wie in Π definiert
- **Transitionen:** $s \xrightarrow{a} s'$ für Zustände s, s' und Aktion a gdw.:
 - partielle Belegung $\text{pre}(a)$ ist Teilbelegung von s (Vorbedingungen erfüllt)
 - s' entspricht $\text{eff}(a)$ für alle Variablen, die in eff erwähnt werden; entspricht s für alle anderen Variablen (Effekte werden angewandt)
- **Anfangszustand:** $s_0 = I$
- **Zielzustände:** $s \in S_*$ für Zustand s gdw. G Teilbelegung von s

Beispiel: Logistikaufgabe mit einem Paket, zwei Lastwagen

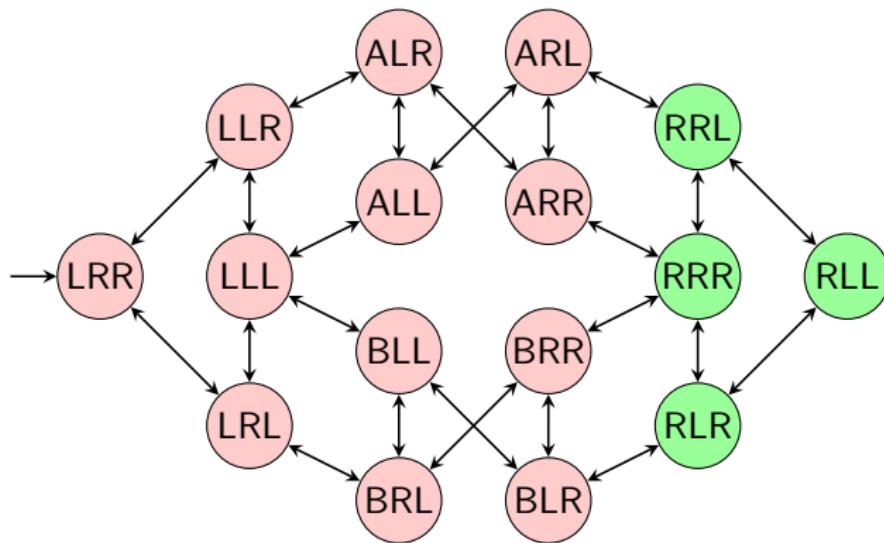
Beispiel (ein Paket, zwei Lastwagen)

Betrachte die SAS^+ -Planungsaufgabe $\langle V, \text{dom}, I, G, A \rangle$ mit:

- $V = \{p, t_A, t_B\}$
- $\text{dom}(p) = \{L, R, A, B\}$ und $\text{dom}(t_A) = \text{dom}(t_B) = \{L, R\}$
- $I = \{p \mapsto L, t_A \mapsto R, t_B \mapsto R\}$ und $G = \{p \mapsto R\}$
- $A = \{pickup_{i,j} \mid i \in \{A, B\}, j \in \{L, R\}\}$
 $\cup \{drop_{i,j} \mid i \in \{A, B\}, j \in \{L, R\}\}$
 $\cup \{move_{i,j,j'} \mid i \in \{A, B\}, j, j' \in \{L, R\}, j \neq j'\}$ mit:
 - $pickup_{i,j}$ hat Vorbedingungen $\{t_i \mapsto j, p \mapsto j\}$, Effekte $\{p \mapsto i\}$
 - $drop_{i,j}$ hat Vorbedingungen $\{t_i \mapsto j, p \mapsto i\}$, Effekte $\{p \mapsto j\}$
 - $move_{i,j,j'}$ hat Vorbedingungen $\{t_i \mapsto j\}$, Effekte $\{t_i \mapsto j'\}$
 - Alle Aktionen haben Kosten 1.

pickup entspricht load und drop entspricht unload aus Kapitel 32 (umbenannt, damit Abkürzungen im Folgenden eindeutiger sind)

Zustandsraum für Beispielaufgabe



- Zustand $\{p \mapsto i, t_A \mapsto j, t_B \mapsto k\}$ abgebildet als ijk .
- Kantenbeschriftungen der Übersicht halber weggelassen.
Zum Beispiel hat die Kante von LLL zu ALL die Beschriftung $pickup_{A,L}$.

SAS+
oooooo

Abstraktionen
●ooooooooo

Musterdatenbanken
oooooooo

Zusammenfassung
oo

Abstraktionen

Abstraktion eines Zustandsraums

Eine Abstraktion eines Zustandsraums **gibt die Unterscheidung zwischen bestimmten Zuständen auf**, bewahrt dabei aber das **Verhalten des Zustandsraums** so weit wie möglich.

- Eine Abstraktion eines Zustandsraums \mathcal{S} ist durch eine **Abstraktionsfunktion α** definiert, die festlegt, welche Zustände unterschieden werden sollen und welche nicht.
- Aus \mathcal{S} und α berechnen wir den **abstrakten Zustandsraum \mathcal{S}^α** , der ähnlich zu \mathcal{S} ist, aber kleiner.

Abstraktionsheuristik

Verwende die **abstrakten Zielabstände** (Zielabstände in \mathcal{S}^α) als Heuristikwerte für die konkreten Zielabstände (Zielabstände in \mathcal{S})
~~> **Abstraktionsheuristik h^α**

Induzierte Abstraktion

Definition (induzierte Abstraktion)

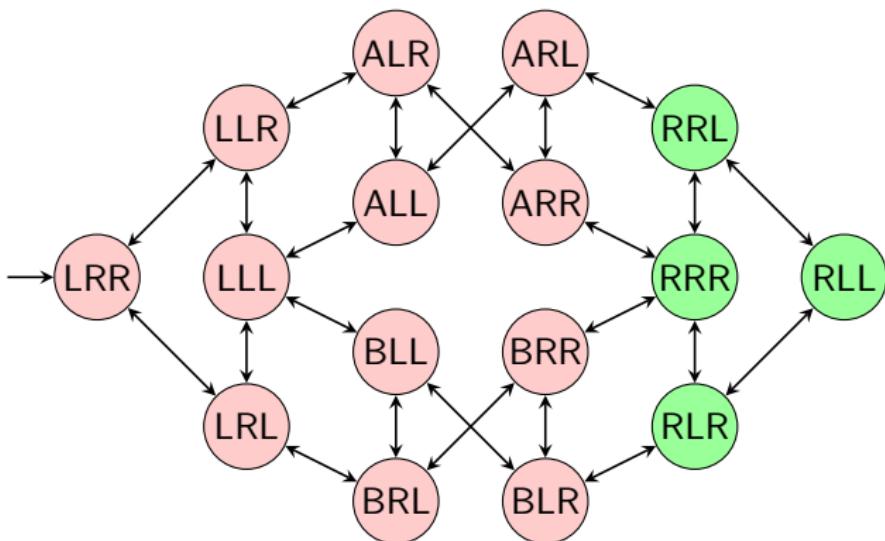
Sei $\mathcal{S} = \langle S, A, \text{cost}, T, s_0, S_* \rangle$ ein Zustandsraum,
und sei $\alpha : S \rightarrow S'$ eine surjektive Funktion.

Die **durch α induzierte Abstraktion von \mathcal{S}** , geschrieben \mathcal{S}^α ,
ist der Zustandsraum $\mathcal{S}^\alpha = \langle S', A, \text{cost}, T', s'_0, S'_* \rangle$ mit:

- $T' = \{\langle \alpha(s), a, \alpha(t) \rangle \mid \langle s, a, t \rangle \in T\}$
- $s'_0 = \alpha(s_0)$
- $S'_* = \{\alpha(s) \mid s \in S_*\}$

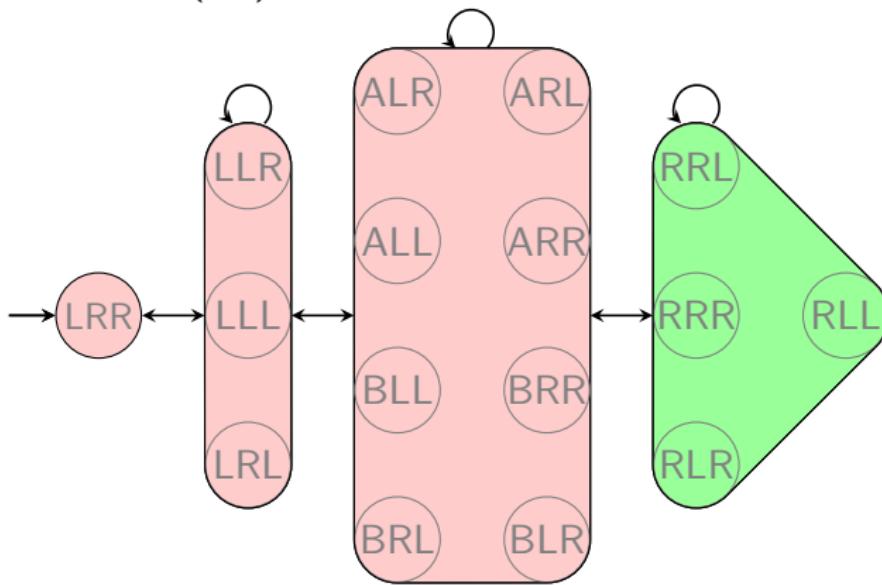
Abstraktion: Beispiel

konkreter Zustandsraum



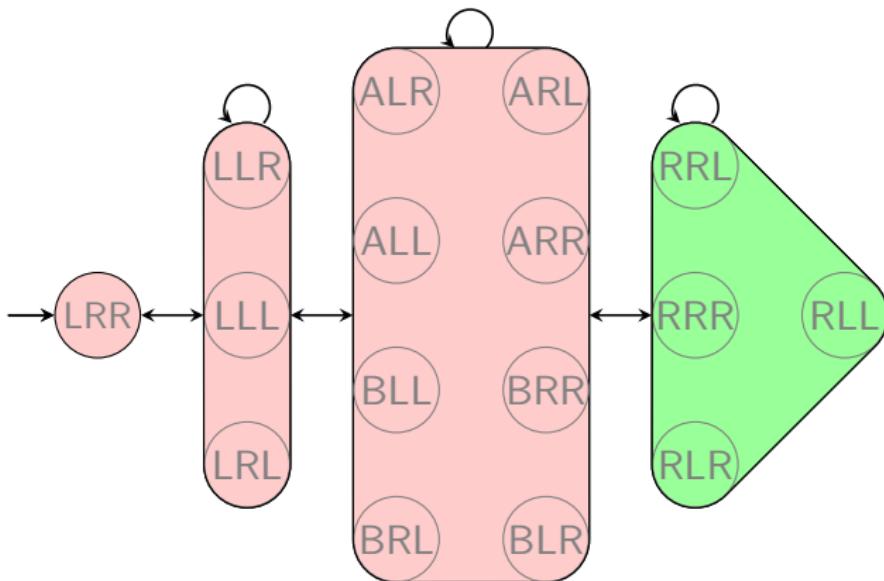
Abstraktion: Beispiel

(ein) abstrakter Zustandsraum



Anmerkung: Die meisten Kanten entsprechen mehreren parallelen Transitionen mit unterschiedlichen Beschriftungen.

Abstraktionsheuristik: Beispiel

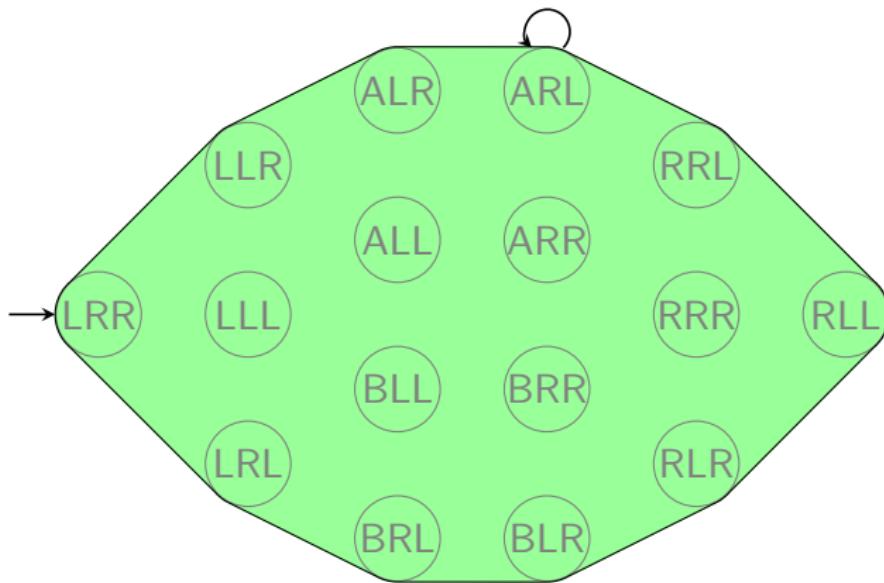


$$h^\alpha(\{p \mapsto L, t_A \mapsto R, t_B \mapsto R\}) = 3$$

Abstraktionsheuristiken: Diskussion

- Jede Abstraktionsheuristik ist **zulässig** und **konsistent**.
(Beweisidee?)
- Die Wahl der **Abstraktionsfunktion α** ist äusserst wichtig.
 - **Jedes α** liefert eine zulässige und konsistente Heuristik.
 - Aber nur wenige solche Heuristiken sind wirklich informativ.
- Ein praktisches α muss eine **informative Heuristik** liefern...
- ... und **effizient berechnet** werden können.
- **Wie finden wir ein geeignetes α ?**

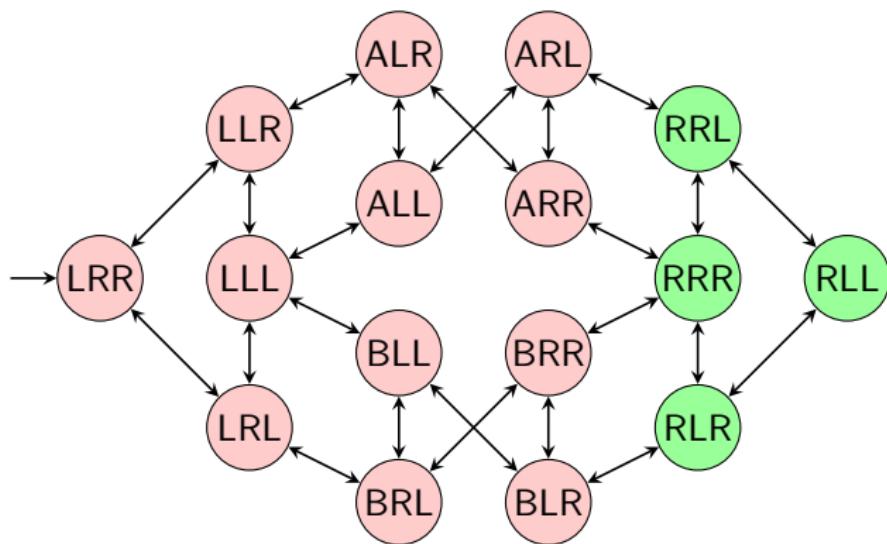
Meist schlechte Idee: Ein-Zustand-Abstraktion



Ein-Zustand-Abstraktion: $\alpha(s) := \text{const}$

- + sehr kompakt repräsentierbar und α leicht berechenbar
- völlig uninformativer Heuristik

Meist schlechte Idee: Identitätsabstraktion



Identitätsabstraktion: $\alpha(s) := s$

- + perfekte Heuristik und α leicht berechenbar
- zu viele abstrakte Zustände \rightsquigarrow Berechnung von h^α zu schwer

Automatische Berechnung guter Abstraktion

Hauptproblem bei Planen mit Abstraktionsheuristiken

Wie finden wir eine gute Abstraktion?

Wir stellen zwei erfolgreiche Methoden vor:

- **Musterdatenbanken** (pattern databases, **PDBs**)
(Culberson & Schaeffer, 1996)
- **Merge-and-Shrink-Abstraktionen**
(Dräger, Finkbeiner & Podelski, 2006)

SAS+
oooooo

Abstraktionen
oooooooooooo

Musterdatenbanken
●oooooooo

Zusammenfassung
oo

Musterdatenbanken

Musterdatenbanken

- Die am häufigsten verwendeten Abstraktionsheuristiken sind **Musterdatenbank-Heuristiken (PDB-Heuristiken)**.
- eingeführt für das **15-Puzzle** (Culberson & Schaeffer, 1996) und den **Zauberwürfel** (Korf, 1997).
- für **Handlungsplanung** eingeführt von Edelkamp (2001)
- für viele Suchprobleme die **besten bekannten** Heuristiken
- viele viele Arbeiten zu
 - theoretischen Eigenschaften
 - effizienter Nutzung
 - Auswahl guter Muster
 - ...

Musterdatenbanken: informell

Musterdatenbanken: informell

Eine PDB-Heuristik für eine Planungsaufgabe ist eine Abstraktionsheuristik, bei der

- einige Aspekte (= Zustandsvariablen) **mit perfekter Genauigkeit** berücksichtigt werden, während
- alle anderen Aspekte **überhaupt nicht** berücksichtigt werden.

Beispiel (15-Puzzle)

- Wähle Teilmenge P der Kacheln (das **Muster** bzw. Pattern).
- Berücksichtige in der Abstraktion den exakten Ort der Kacheln in P .
- Nimm in der Abstraktion an, dass alle anderen Kacheln und das freie Feld überall sein können.

Projektionen

PDB-Heuristiken sind Abstraktionsheuristiken für eine bestimmte Art von Abstraktionsfunktionen, nämlich **Projektionen**.

Definition (Projektion)

Sei Π eine Planungsaufgabe mit Variablen V und Zuständen S .

Sei $P \subseteq V$, und sei S' die Menge der partiellen Belegungen, die genau auf P definiert sind.

Die **Projektion** $\pi_P : S \rightarrow S'$ ist definiert als $\pi_P(s) := s|_P$ (mit $s|_P(v) := s(v)$ für alle $v \in P$).

Wir nennen P das **Muster (Pattern)** der Projektion π_P .

Anders gesagt: π_P bildet zwei Zustände s_1 und s_2 genau dann auf denselben abstrakten Zustand ab, wenn sie auf allen Variablen in P übereinstimmen.

Musterdatenbank-Heuristiken

Definition (Musterdatenbank-Heuristik)

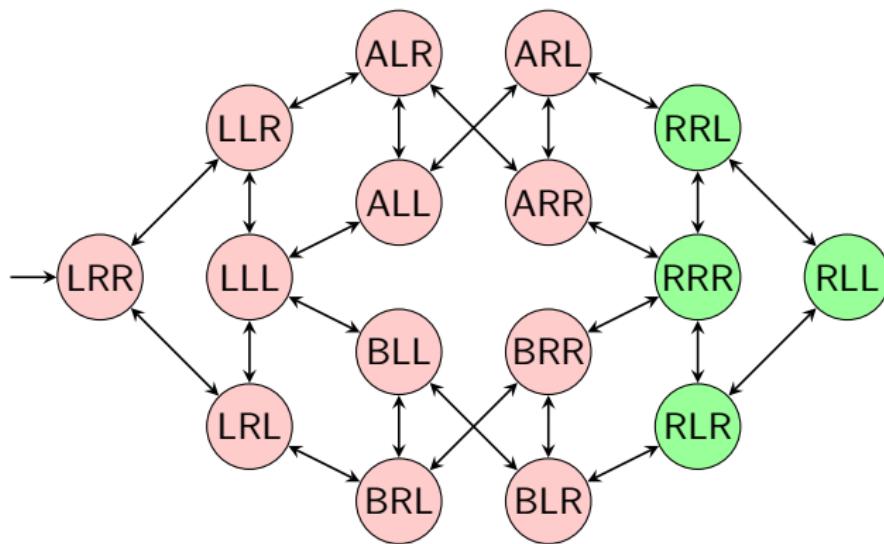
Abstraktionsheuristiken, die durch Projektion π_P induziert werden, heißen **Musterdatenbank-Heuristiken (PDB-Heuristiken)**.

Kurzschreibweise: h^P für h^{π_P}

Warum der Name PDB-Heuristik?

- Heuristikwerte für PDB-Heuristiken werden üblicherweise in einer 1-dimensionalen Tabelle (Array) gespeichert.
- Diese Tabellen nennt man PDBs.

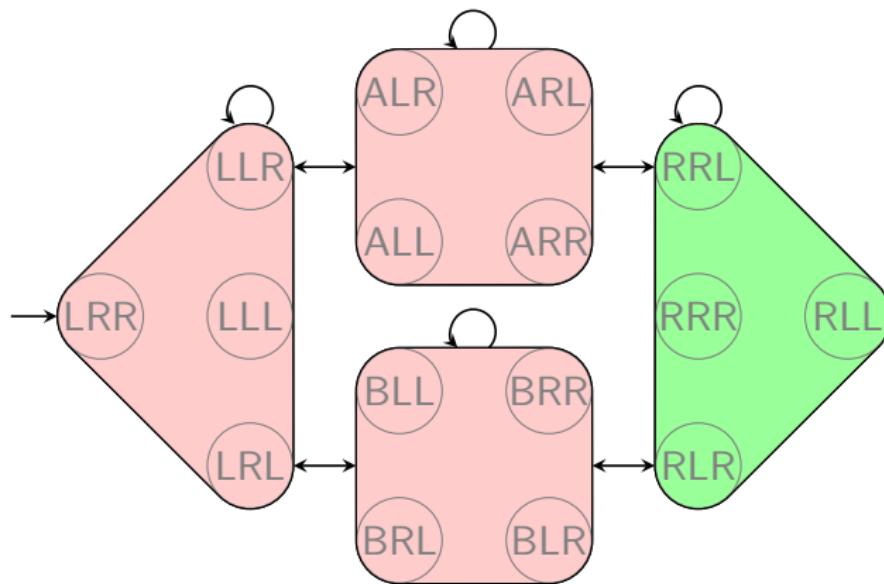
Beispiel: konkreter Zustandsraum



- Zustandsvariable *package*: {L, R, A, B}
- Zustandsvariable *truck A*: {L, R}
- Zustandsvariable *truck B*: {L, R}

Beispiel: Projektion (1)

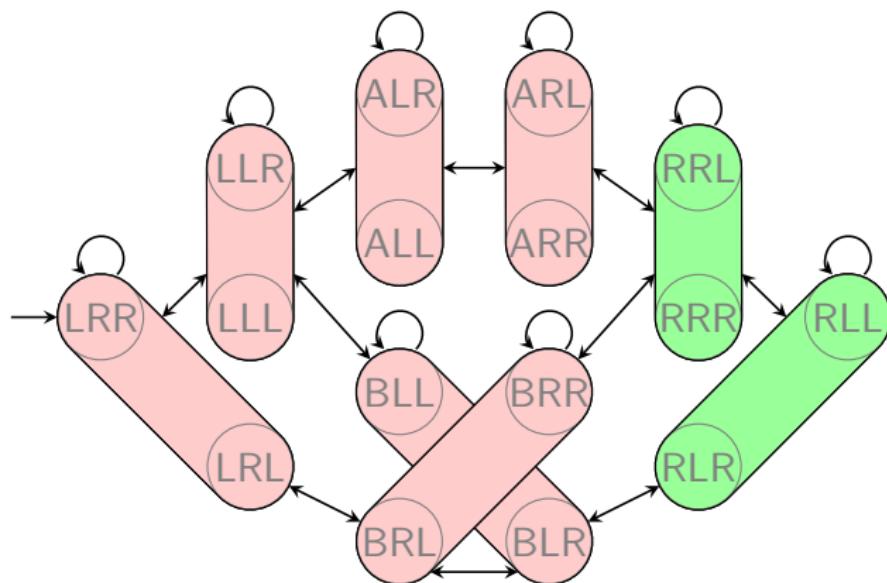
Von $\pi_{\{package\}}$ induzierte Abstraktion:



$$h^{\{package\}}(\text{LRR}) = 2$$

Beispiel: Projektion (2)

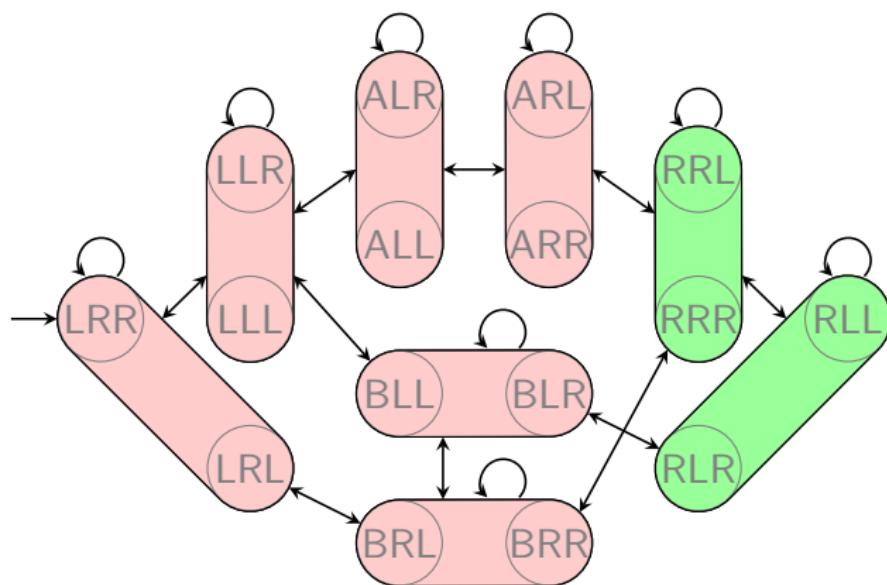
Von $\pi_{\{\text{package}, \text{truck A}\}}$ induzierte Abstraktion:



$$h^{\{\text{package}, \text{truck A}\}}(\text{LRR}) = 2$$

Beispiel: Projektion (2)

Von $\pi_{\{package, truck A\}}$ induzierte Abstraktion:



$$h^{\{package, truck A\}}(\text{LRR}) = 2$$

Musterdatenbanken in der Praxis

Praktische Aspekte, auf die wir nicht eingehen:

- Wie finden wir **automatisch gute Patterns**?
- Wie kombinieren wir sinnvoll **mehrere PDB-Heuristiken**?
- Wie **implementieren** wir PDB-Heuristiken effizient?
 - gute Implementierungen berechnen schnell Patterns für **abstrakte Zustandsräume** mit 10^7 , 10^8 oder mehr Zuständen
 - Aufwand unabhängig von Grösse des **konkreten Zustandsraums**
 - meist werden alle Heuristikwerte vorberechnet
~~ Platzaufwand = Anzahl abstrakter Zustände

SAS+
oooooo

Abstraktionen
oooooooooooo

Musterdatenbanken
oooooooooooo

Zusammenfassung
●○

Zusammenfassung

Zusammenfassung

- Grundidee **Abstraktionsheuristiken**: Schätze Lösungskosten durch Betrachten einer **kleineren** Planungsaufgabe ab.
- formal: **Abstraktionsfunktion** α bildet Zustände auf **abstrakte Zustände** ab und definiert so, welche Zustände von der Heuristik unterschieden werden und welche nicht.
- induziert **abstrakten Zustandsraum**, dessen Zielabstände als Heuristik verwendet werden
- **Musterdatenbank-Heuristiken** sind Abstraktionsheuristiken, die auf **Projektion** auf Teilmenge der Zustandsvariablen (**Muster**) basieren: Zustände werden unterschieden, wenn sie sich auf dem Muster unterscheiden.