

Grundlagen der Künstlichen Intelligenz

16. Handlungsplanung: Abstraktion

Malte Helmert

Universität Basel

13. Mai 2013

Grundlagen der Künstlichen Intelligenz

13. Mai 2013 — 16. Handlungsplanung: Abstraktion

16.1 SAS⁺

16.2 Abstraktionen

16.3 Musterdatenbanken

16.4 Merge-and-Shrink

Handlungsplanung: Überblick

Kapitelüberblick:

- ▶ Einführung und Formalisierung $\rightsquigarrow$ Kapitel 14
- ▶ Drei Kernideen für allgemeine Heuristiken:
 - ▶ Delete-Relaxierung $\rightsquigarrow$ Kapitel 15
 - ▶ **Abstraktion** $\rightsquigarrow$ dieses Kapitel
 - ▶ Landmarken $\rightsquigarrow$ Kapitel 17

Grundidee Abstraktion

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

16.1 SAS⁺

SAS⁺-Kodierung

- ▶ In diesem Kapitel: **SAS⁺**-Kodierung statt STRIPS (siehe Kapitel 14)
- ▶ 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** $\mathcal{S}(\Pi) = \langle S, A, \text{cost}, T, s_0, S_\star \rangle$:

- ▶ **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**: $s \in S_\star$ für Zustand s gdw. G Teilbelegung von s

Beispiel: Logistikaufgabe mit einem Paket, zwei Lastwagen

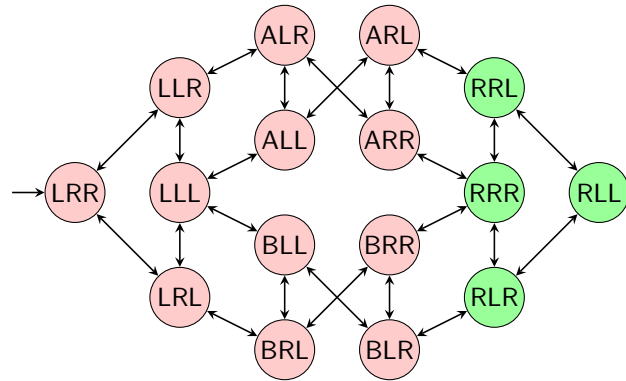
Beispiel (ein Paket, zwei Lastwagen)

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 $G = \{p \mapsto R\}$
- ▶ $A = \{ \text{pickup}_{i,j} \mid i \in \{A, B\}, j \in \{L, R\} \}$
 $\cup \{ \text{drop}_{i,j} \mid i \in \{A, B\}, j \in \{L, R\} \}$
 $\cup \{ \text{move}_{i,j,j'} \mid i \in \{A, B\}, j, j' \in \{L, R\}, j \neq j' \}$ mit:
 - ▶ $\text{pickup}_{i,j}$ hat Vorbedingungen $\{t_i \mapsto j, p \mapsto j\}$, Effekte $\{p \mapsto i\}$
 - ▶ $\text{drop}_{i,j}$ hat Vorbedingungen $\{t_i \mapsto j, p \mapsto i\}$, Effekte $\{p \mapsto j\}$
 - ▶ $\text{move}_{i,j,j'}$ hat Vorbedingungen $\{t_i \mapsto j\}$, Effekte $\{t_i \mapsto j'\}$
 - ▶ Alle Aktionen haben Kosten 1.

pickup entspricht **load** und **drop** entspricht **unload** im Vorkapitel (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 $pickup_{A,L}$.

16.2 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** $\mathcal{S}'$, der ähnlich zu $\mathcal{S}$ ist, aber kleiner.

Abstraktionsheuristik

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

Induzierte Abstraktion

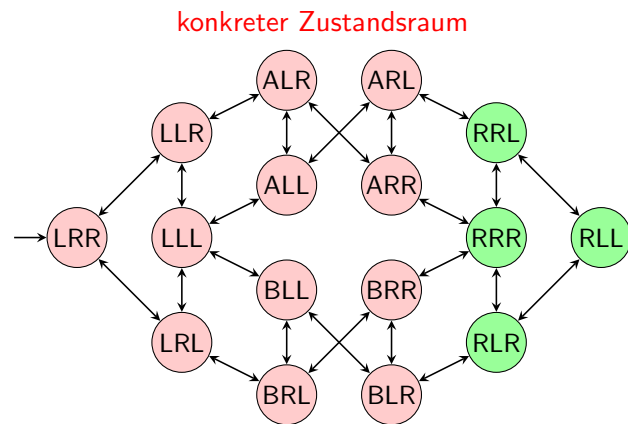
Definition (induzierte Abstraktion)

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

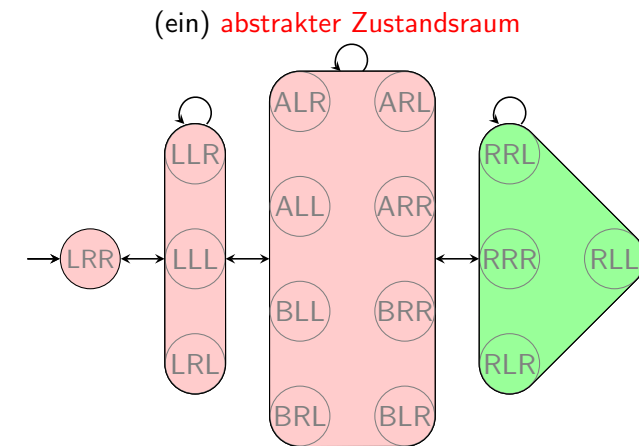
Die **durch α induzierte Abstraktion von $\mathcal{S}$** , geschrieben $\mathcal{S}^\alpha$, ist der Zustandsraum $\mathcal{S}^\alpha = \langle S', A, cost, T', s'_0, S'_\star \rangle$ mit:

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

Abstraktion: Beispiel

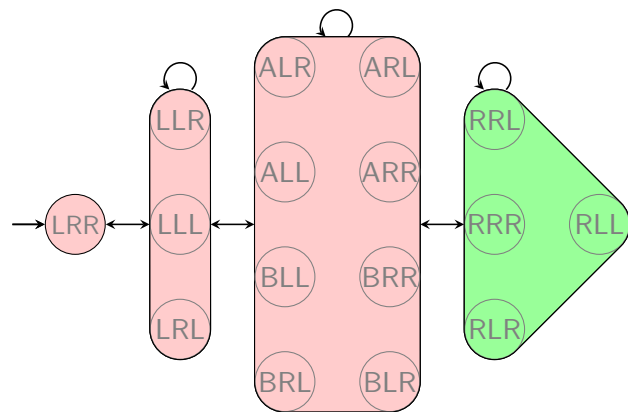


Abstraktion: Beispiel



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

Abstraktionsheuristik: Beispiel

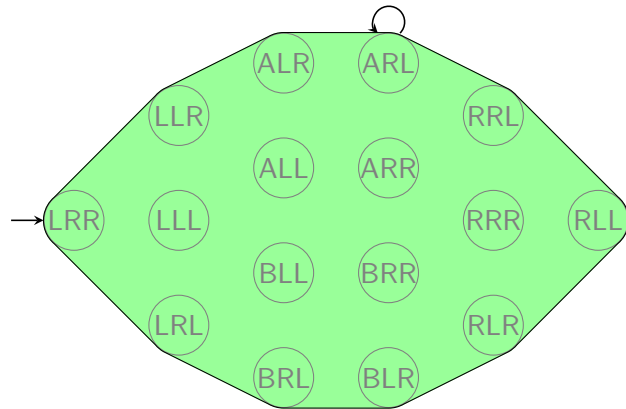


$$h^\alpha(\{p \mapsto L, t_A \mapsto R, t_B \mapsto R\}) = 3$$

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 ?**

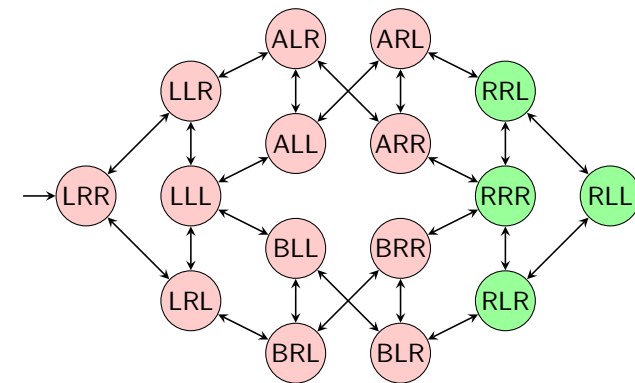
Meist schlechte Idee: Ein-Zustand-Abstraktion



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

- + sehr kompakt repräsentierbar und α leicht berechenbar
- völlig uninformative 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)

16.3 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)

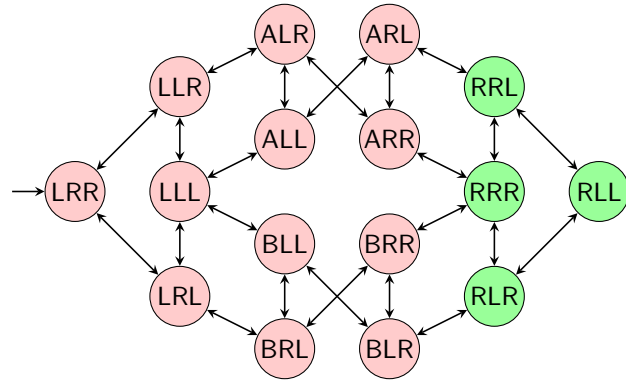
Abstraktionsheuristiken, die durch Projektion π_P induziert werden, heißen **Musterdatenbank-Heuristiken** (**PDB-Heuristiken**).

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.

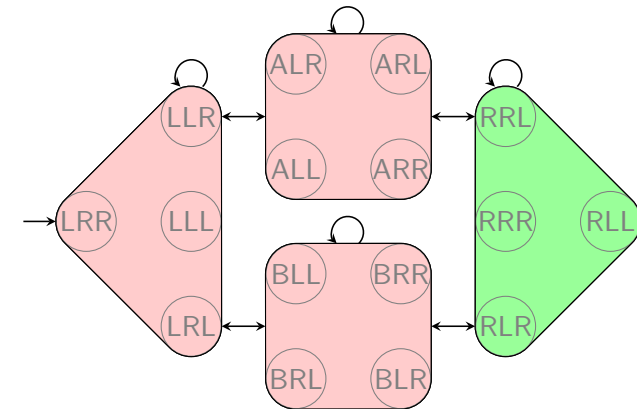
Beispiel: konkreter Zustandsraum



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

Beispiel: Projektion (1)

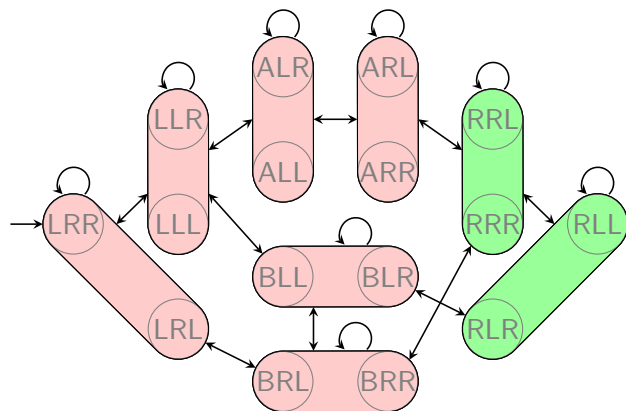
Von $\pi_{\{package\}}$ induzierte Abstraktion:



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

Beispiel: Projektion (2)

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



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

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
 $\rightsquigarrow$ Platzaufwand = Anzahl abstrakter Zustände

16.4 Merge-and-Shrink

Jenseits von Musterdatenbanken

- ▶ Trotz ihrer Popularität haben PDBs fundamentale Grenzen:
die Muster müssen klein gehalten werden
- ▶ Preis in heuristischer Genauigkeit:
 - ▶ betrachte Verallgemeinerung des Beispiels:
 N Lastwagen, M Orte (ein Paket)
 - ▶ betrachte **beliebiges** Muster, das nicht alle Variablen in V umfasst
 - ▶ $h(s_0) \leq 2 \rightsquigarrow$ **nicht besser** als atomare Projektion auf das Paket
- ▶ **Merge-and-Shrink-Abstraktionen (M&S)** sind eine **echte Verallgemeinerung** von PDBs.
 - ▶ Sie können PDBs repräsentieren (mit polynomiellm Zusatzaufwand).
 - ▶ Sie können Abstraktionen kompakt repräsentieren, bei denen dies mit PDBs nicht möglich ist.

Merge-and-Shrink: Abgrenzung zu PDBs

M&S-Abstraktionen vs. Musterdatenbanken

Während PDBs **einige wenige** Zustandsvariablen **perfekt** im abstrakten Zustandsraum repräsentieren, repräsentieren M&S-Abstraktionen **alle** Zustandsvariablen, aber in einer **verlustbehafteten** Weise.

M&S-Abstraktionen: Kernideen

Kernideen bei M&S:

- ① Informationen von zwei abstrakten Zustandsräumen $\mathcal{A}$ und $\mathcal{A}'$ für denselben konkreten Zustandsraum können durch eine einfache Graphenoperation **kombiniert** werden:
synchrones Produkt $\mathcal{A} \otimes \mathcal{A}'$.
- ② Der **konkrete** Zustandsraum $\mathcal{S}$ einer SAS⁺-Planungsaufgabe kann aus den **atomaren Projektionen** rekonstruiert werden:

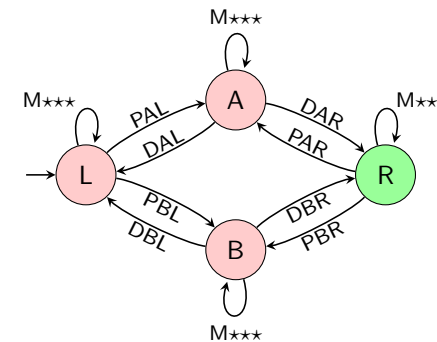
$$\bigotimes_{v \in V} \mathcal{S}^{\pi\{v\}} \text{ ist isomorph zu } \mathcal{S}$$

$\rightsquigarrow$ baue feine Abstraktionen aus gröberen

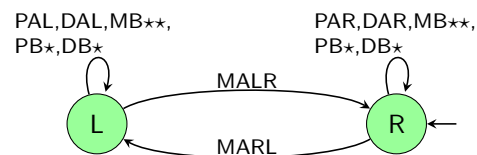
Beispiel: Abkürzungen

- Für das synchrone Produkt sind die **Kantenbeschriftungen** in den Zustandsräumen (die „Aktionsnamen“) sehr wichtig.
- Wir geben sie deshalb im Folgenden in den Abbildungen an.
- Dabei verwenden wir Abkürzungen der folgenden Art:
 - **MALR**: move truck **A** from **left** to **right**
 - **DAR**: drop package from truck **A** at **right** location
 - **PBL**: pick up package with truck **B** at **left** location
- Oft gibt es viele Parallelkanten. Wir kürzen diese mit **Kommata** und **Platzhaltern** ab wie in diesen Beispielen:
 - **PAL, DAL**: parallele Kanten für Aktionen **PAL** und **DAL**
 - **MA****: parallele Kanten für Aktionen **MALR** und **MARL**

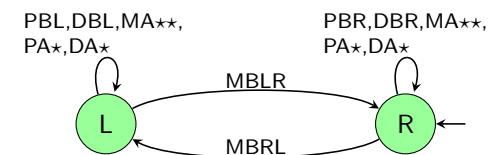
Beispiel: atomare Projektion für das Paket

 $\mathcal{S}^{\pi}_{\{\text{package}\}}:$


Beispiel: atomare Projektion für Lastwagen A

 $\mathcal{S}^{\pi}_{\{\text{truck A}\}}:$


Beispiel: atomare Projektion für Lastwagen B

 $\mathcal{S}^{\pi}_{\{\text{truck B}\}}:$


Synchrones Produkt von Zustandsräumen

Definition (synchrones Produkt von Zustandsräumen)

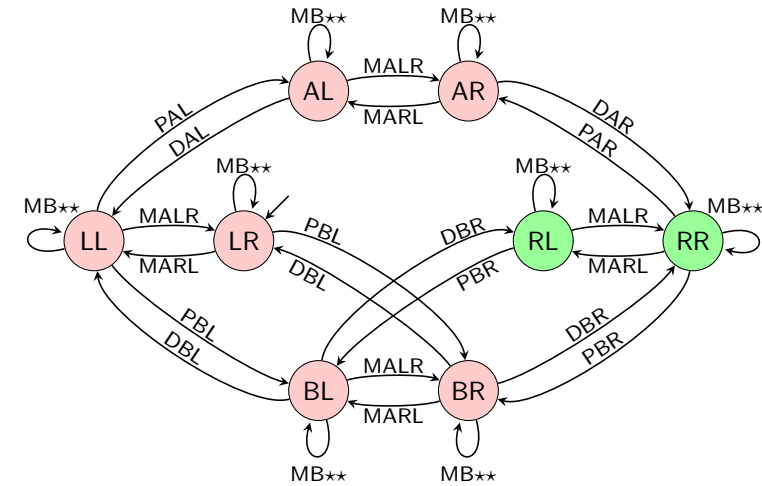
Für $i \in \{1, 2\}$ seien $\mathcal{S}^i = \langle S^i, A, cost, T^i, s_0^i, S_\star^i \rangle$ Zustandsräume (mit identischen Aktionen und Kosten).

Das **synchrone Produkt** von $\mathcal{S}^1$ und $\mathcal{S}^2$, geschrieben $\mathcal{S}^1 \otimes \mathcal{S}^2$, ist der Zustandsraum $\mathcal{S}^\otimes = \langle S^\otimes, A, cost, T^\otimes, s_0^\otimes, S_\star^\otimes \rangle$ mit

- ▶ $S^\otimes := S^1 \times S^2$
- ▶ $T^\otimes := \{ \langle \langle s_1, s_2 \rangle, a, \langle t_1, t_2 \rangle \rangle \mid \langle s_1, a, t_1 \rangle \in T^1 \wedge \langle s_2, a, t_2 \rangle \in T^2 \}$
- ▶ $s_0^\otimes := \langle s_0^1, s_0^2 \rangle$
- ▶ $S_\star^\otimes := S_\star^1 \times S_\star^2$

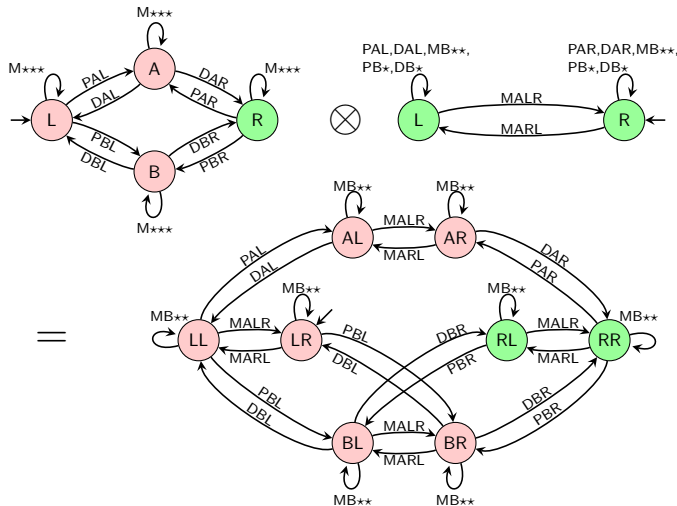
Beispiel: synchrones Produkt

$$\mathcal{S}^\pi\{\text{package}\} \otimes \mathcal{S}^\pi\{\text{truck } A\}:$$



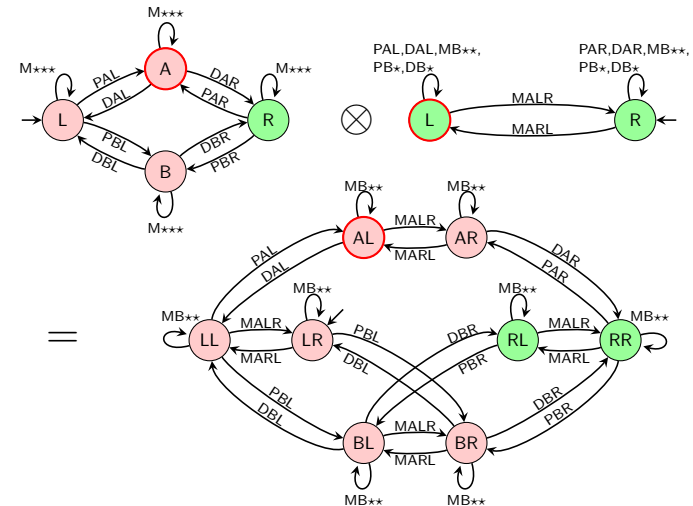
Beispiel: Berechnung des synchronen Produkts

$$\mathcal{S}^\pi\{\text{package}\} \otimes \mathcal{S}^\pi\{\text{truck } A\}:$$



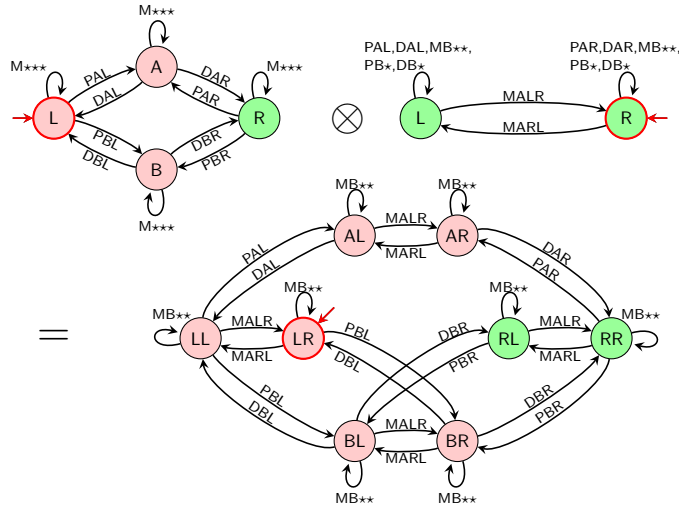
Beispiel: Berechnung des synchronen Produkts

$$\mathcal{S}^\pi\{\text{package}\} \otimes \mathcal{S}^\pi\{\text{truck } A\}: \mathcal{S}^\otimes = \mathcal{S}^1 \times \mathcal{S}^2$$



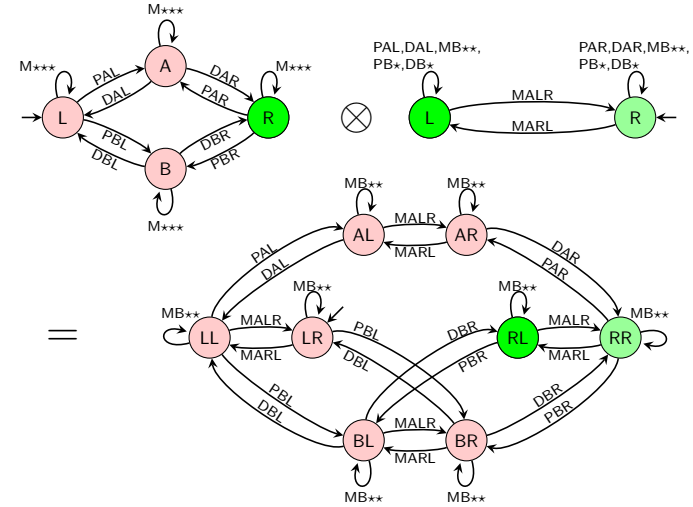
Beispiel: Berechnung des synchronen Produkts

$$\mathcal{S}^{\pi}_{\{package\}} \otimes \mathcal{S}^{\pi}_{\{truck\} A}: s_0^{\otimes} = \langle s_0^1, s_0^2 \rangle$$



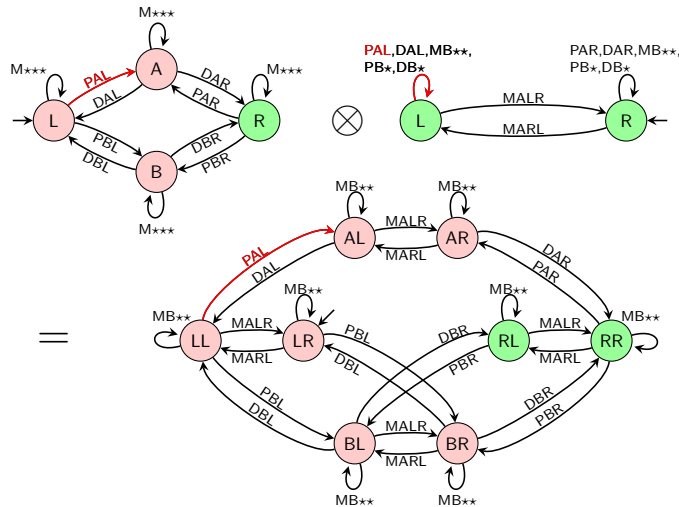
Beispiel: Berechnung des synchronen Produkts

$$\mathcal{S}^{\pi}_{\{package\}} \otimes \mathcal{S}^{\pi}_{\{truck\} A}: s_{\star}^{\otimes} = s_{\star}^1 \times s_{\star}^2$$



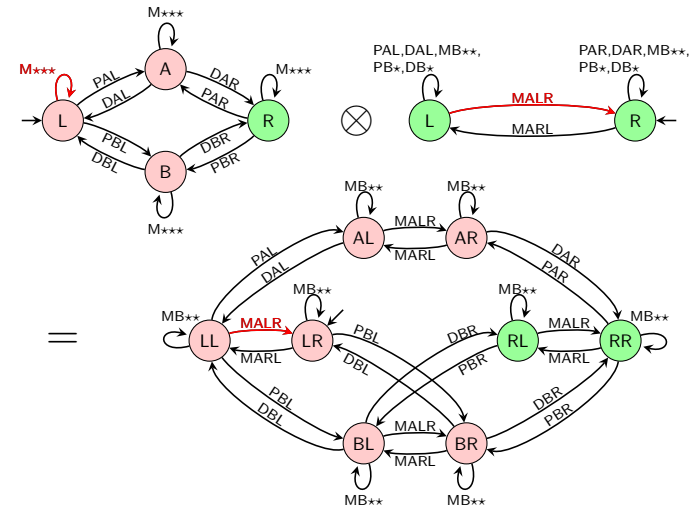
Beispiel: Berechnung des synchronen Produkts

$$\mathcal{S}^{\pi}_{\{package\}} \otimes \mathcal{S}^{\pi}_{\{truck\} A}: T^{\otimes} = \{ \langle \langle s_1, s_2 \rangle, a, \langle t_1, t_2 \rangle \rangle \mid \dots \}$$



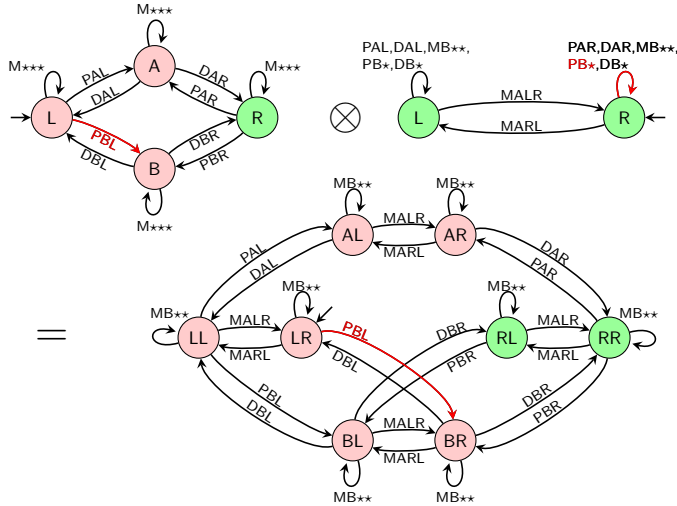
Beispiel: Berechnung des synchronen Produkts

$$\mathcal{S}^{\pi}_{\{package\}} \otimes \mathcal{S}^{\pi}_{\{truck\} A}: T^{\otimes} = \{ \langle \langle s_1, s_2 \rangle, a, \langle t_1, t_2 \rangle \rangle \mid \dots \}$$



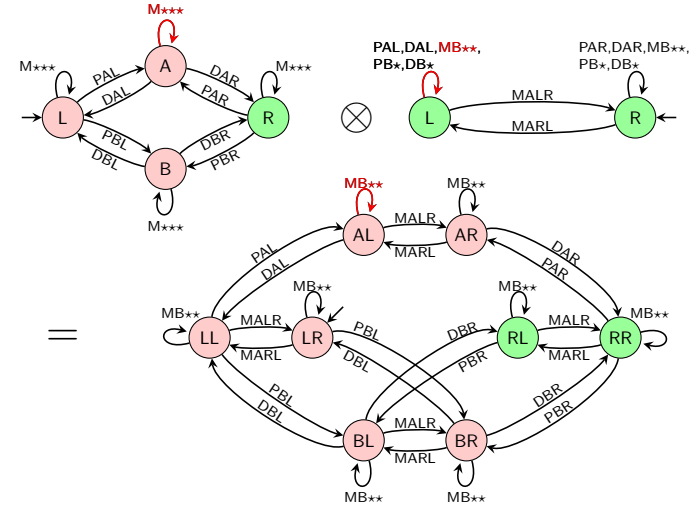
Beispiel: Berechnung des synchronen Produkts

$$\mathcal{S}^{\pi}_{\{\text{package}\}} \otimes \mathcal{S}^{\pi}_{\{\text{truck } A\}}: T^{\otimes} = \{ \langle \langle s_1, s_2 \rangle, a, \langle t_1, t_2 \rangle \rangle \mid \dots \}$$



Beispiel: Berechnung des synchronen Produkts

$$\mathcal{S}^{\pi}_{\{\text{package}\}} \otimes \mathcal{S}^{\pi}_{\{\text{truck } A\}}: T^{\otimes} = \{ \langle \langle s_1, s_2 \rangle, a, \langle t_1, t_2 \rangle \rangle \mid \dots \}$$



M&S-Abstraktionen: Kernideen (Fortsetzung)

Kernideen bei M&S:

- 1 Information von zwei abstrakten Zustandsräumen $\mathcal{A}$ und $\mathcal{A}'$ für denselben konkreten Zustandsraum können durch eine einfache Graphenoperation **kombiniert** werden: **synchrones Produkt** $\mathcal{A} \otimes \mathcal{A}'$.
- 2 Der **konkrete** Zustandsraum $\mathcal{S}$ einer SAS⁺-Planungsaufgabe kann aus den **atomaren Projektionen** rekonstruiert werden:

$$\bigotimes_{v \in V} \mathcal{S}^{\pi_{\{v\}}} \text{ ist isomorph zu } \mathcal{S}$$

$\rightsquigarrow$ baue feine Abstraktionen aus gröberen

- 3 Wenn Zwischenergebnisse zu gross werden:
Verkleinern durch Kombination einiger abstrakter Zustände

Berechnung von M&S-Abstraktionen

Generischer Algorithmus zur Berechnung von M&S-Abstraktionen

```

abs := {  $\mathcal{S}^{\pi_{\{v\}}}$  |  $v \in V$  } [Abstraktionen für atomare Projektionen]
while |abs| > 1:
  select  $\mathcal{S}^1, \mathcal{S}^2$  from abs
  shrink  $\mathcal{S}^1$  and/or  $\mathcal{S}^2$  until  $\text{size}(\mathcal{S}^1) \cdot \text{size}(\mathcal{S}^2) \leq K$ 
  abs := abs \ {  $\mathcal{S}^1, \mathcal{S}^2$  }  $\cup$  {  $\mathcal{S}^1 \otimes \mathcal{S}^2$  } [Merge-Schritt]
return the remaining abstraction in abs

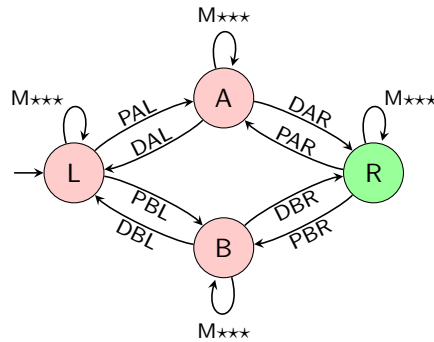
```

K: Parameter, der max. Anzahl abstrakter Zustände begrenzt

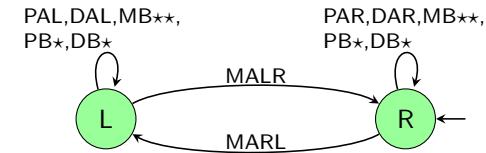
Praktische Implementierungen müssen entscheiden:

- Welche Abstraktionen werden ausgewählt? $\rightsquigarrow$ **Merge-Strategie**
- Wie werden Abstraktionen geschrumpft? $\rightsquigarrow$ **Shrink-Strategie**
- Wie soll K gewählt werden?

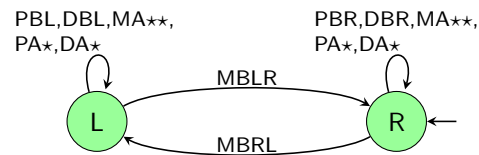
Initialisierungsschritt: atomare Projektion für das Paket

 $\mathcal{S}^\pi_{\{\text{package}\}}:$ 

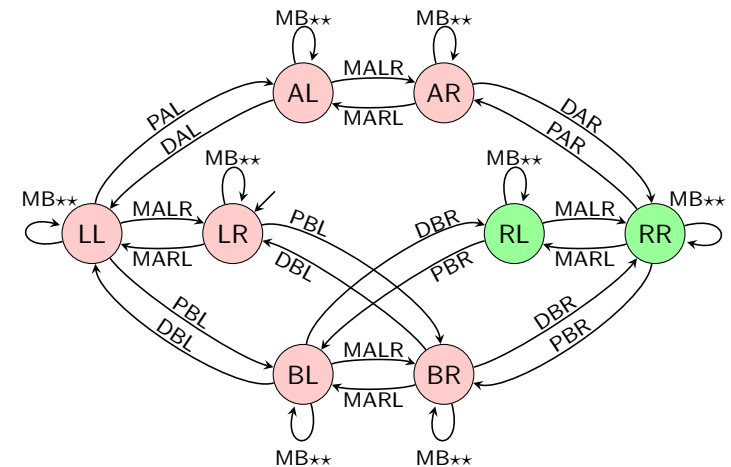
Initialisierungsschritt: atomare Projektion für Lastwagen A

 $\mathcal{S}^\pi_{\{\text{truck A}\}}:$ 

Initialisierungsschritt: atomare Projektion für Lastwagen B

 $\mathcal{S}^\pi_{\{\text{truck B}\}}:$ aktuelle Abstraktionsmenge: $abs = \{\mathcal{S}^\pi_{\{\text{package}\}}, \mathcal{S}^\pi_{\{\text{truck A}\}}, \mathcal{S}^\pi_{\{\text{truck B}\}}\}$

Erster Merge-Schritt

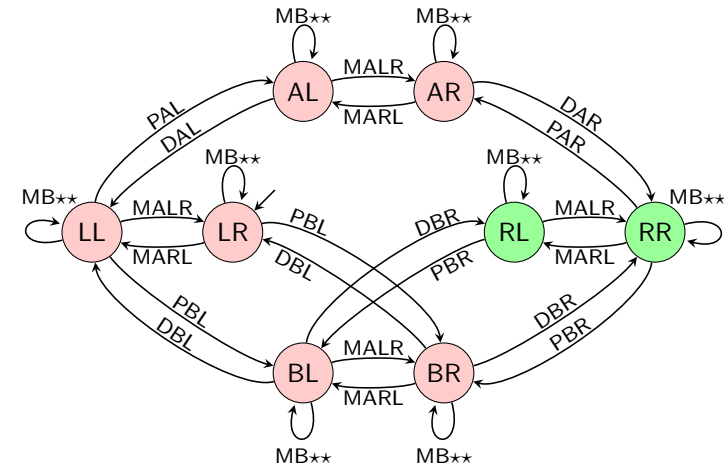
 $\mathcal{S}^1 := \mathcal{S}^\pi_{\{\text{package}\}} \otimes \mathcal{S}^\pi_{\{\text{truck A}\}}:$ aktuelle Abstraktionsmenge: $abs = \{\mathcal{S}^1, \mathcal{S}^\pi_{\{\text{truck B}\}}\}$

Müssen wir vereinfachen?

- ▶ Wenn wir genügend Speicher haben, können wir nun $S^1 \otimes S^{\pi\{truck\ B\}}$ berechnen, wonach wir den konkreten Zustandsraum des Problems konstruiert hätten.
- ▶ Um die allgemeine Idee zu illustrieren, nehmen wir jedoch an, dass wir nicht genug Speicher für dieses Produkt zur Verfügung haben.
- ▶ Genauer: wir nehmen an, dass wir nach jedem Merge-Schritt auf **vier** Zustände reduzieren müssen, um den Speicherverbrauch unter Kontrolle zu halten

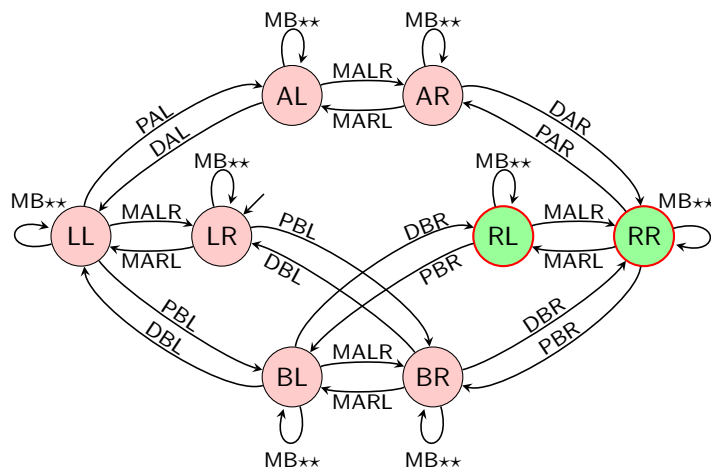
Erster Shrink-Schritt

$S^2 :=$ eine Abstraktion von S^1



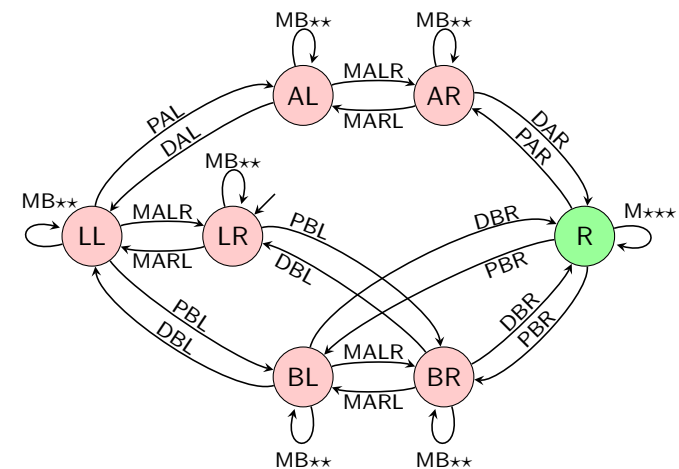
Erster Shrink-Schritt

$S^2 :=$ eine Abstraktion von S^1

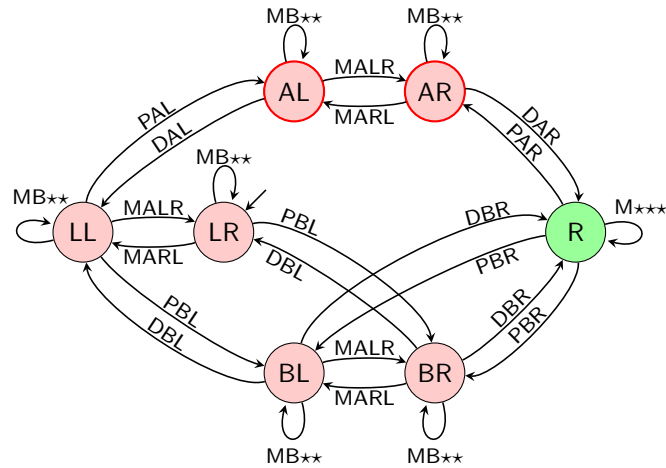


Erster Shrink-Schritt

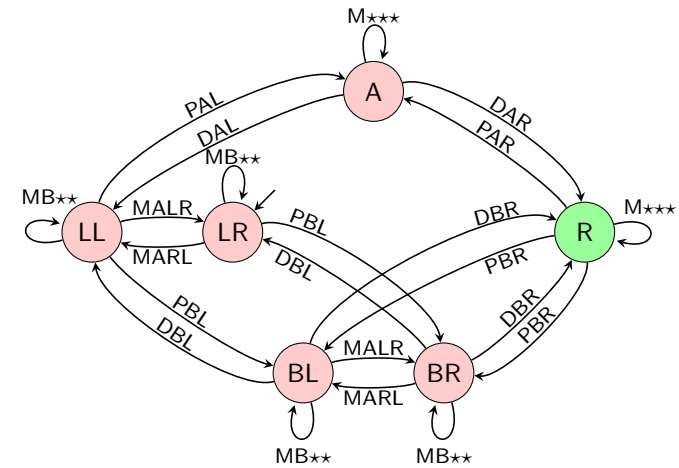
$S^2 :=$ eine Abstraktion von S^1



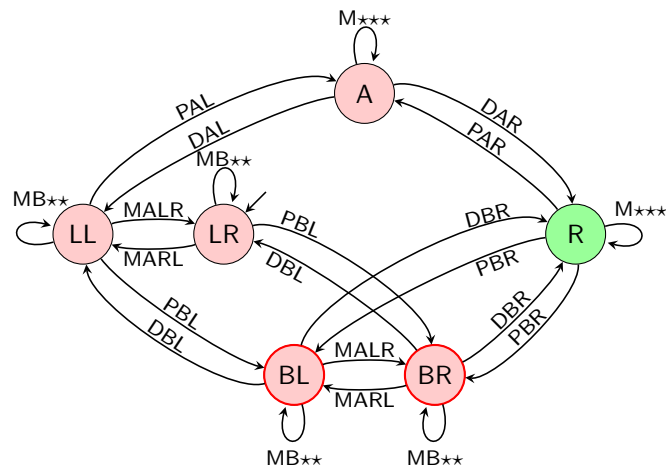
Erster Shrink-Schritt

 $S^2 := \text{eine Abstraktion von } S^1$


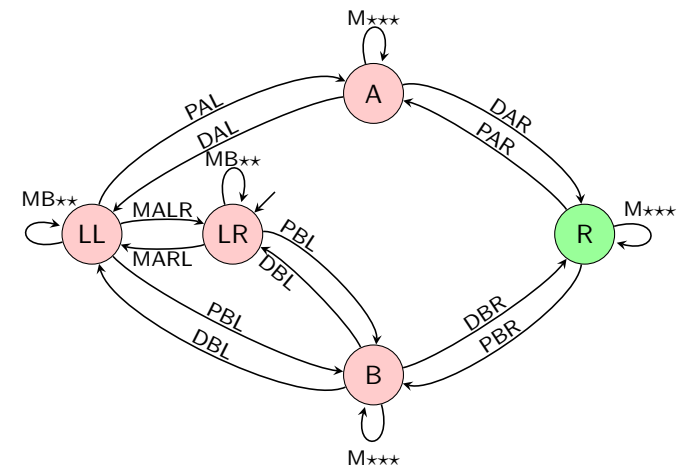
Erster Shrink-Schritt

 $S^2 := \text{eine Abstraktion von } S^1$


Erster Shrink-Schritt

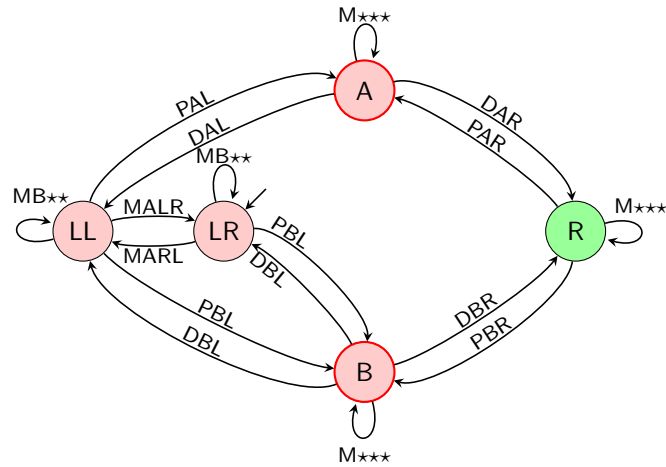
 $S^2 := \text{eine Abstraktion von } S^1$


Erster Shrink-Schritt

 $S^2 := \text{eine Abstraktion von } S^1$


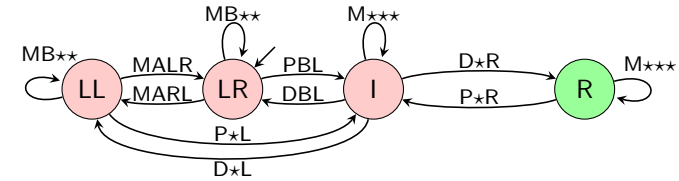
Erster Shrink-Schritt

$\mathcal{S}^2 :=$ eine Abstraktion von $\mathcal{S}^1$



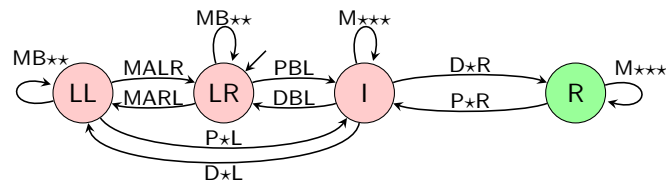
Erster Shrink-Schritt

$\mathcal{S}^2 :=$ eine Abstraktion von $\mathcal{S}^1$



Erster Shrink-Schritt

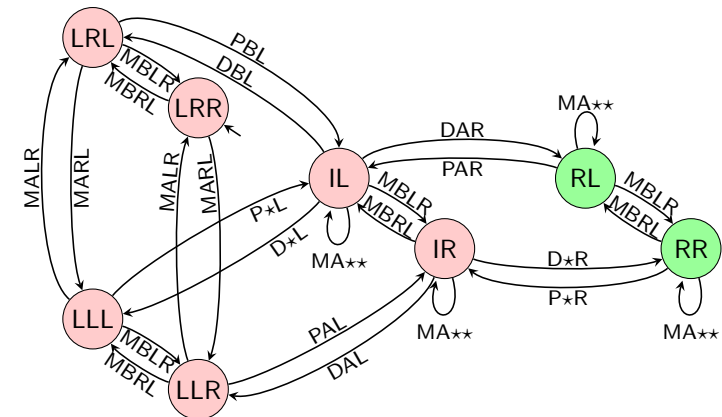
$\mathcal{S}^2 :=$ eine Abstraktion von $\mathcal{S}^1$



aktuelle Abstraktionsmenge: $abs = \{\mathcal{S}^2, \mathcal{S}^{\pi\{truck\ B\}}\}$

Zweiter Merge-Schritt

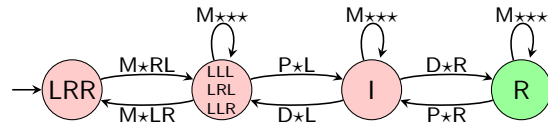
$\mathcal{S}^3 := \mathcal{S}^2 \otimes \mathcal{S}^{\pi\{truck\ B\}} :$



aktuelle Abstraktionsmenge: $\{\mathcal{S}^3\}$

Zweiter Shrink-Schritt

- Wir schrumpfen (um die Ideen zu illustrieren; der generische Algorithmus wäre hier fertig) S^3 noch zu S^4 und erhalten:



- Wir erhalten einen Heuristikwert von 3 für den Anfangszustand $\rightsquigarrow$ **besser als jede PDB-Heuristik**, die nicht alle Variablen im Muster hat.
- Das Beispiel lässt sich auf mehr Orte und Lastwagen verallgemeinern, ohne dass man die Grössenschranke von 4 (nach dem Merge-Schritt) erhöhen müsste.

Merge-and-Shrink-Abstraktionen in der Praxis

Praktische Aspekte, auf die wir nicht eingehen:

- Wie wählen wir den Grössenparameter?
- Welche Merge-Strategien sind gut?
- Welche Shrink-Strategien sind gut?
- Wie **implementieren** wir Merge-and-Shrink effizient?
 - gute Datenstrukturen und Algorithmen wichtig!