

Grundlagen der Künstlichen Intelligenz

38. Handlungsplanung: Merge-and-Shrink-Abstraktionen

Malte Helmert

Universität Basel

18. Mai 2015

Grundlagen der Künstlichen Intelligenz

18. Mai 2015 — 38. Handlungsplanung: Merge-and-Shrink-Abstraktionen

38.1 Merge-and-Shrink: Motivation

38.2 Synchrone Produkt

38.3 Merge-and-Shrink

38.4 Zusammenfassung

Handlungsplanung: Überblick

Kapitelüberblick:

- ▶ 33. Einführung
- ▶ 34. Planungsformalismen
- ▶ 35.–36. Planungsheuristiken: Delete-Relaxierung
- ▶ 37.–38. Planungsheuristiken: Abstraktion
 - ▶ 37. Abstraktion und Musterdatenbanken
 - ▶ 38. Merge-and-Shrink-Abstraktionen
- ▶ 39.–40. Planungsheuristiken: Landmarken

38.1 Merge-and-Shrink: Motivation

Jenseits von Musterdatenbanken (1)

- ▶ Trotz ihrer Popularität haben PDBs fundamentale Grenzen:
die Muster müssen klein gehalten werden
- ▶ Preis in heuristischer Genauigkeit:
 - ▶ betrachte Verallgemeinerung des Beispiels aus dem Vorkapitel:
 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: 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.

Jenseits von Musterdatenbanken (2)

Merge-and-Shrink-Abstraktionen (M&S) sind eine **echte Verallgemeinerung** von PDBs.

- ▶ Sie können PDBs repräsentieren (mit polynomiellem Zusatzaufwand).
- ▶ Sie können Abstraktionen kompakt repräsentieren, bei denen dies mit PDBs nicht möglich ist.

38.2 Synchrones Produkt

M&S-Abstraktionen: Kernideen

Kernideen bei M&S:

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

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

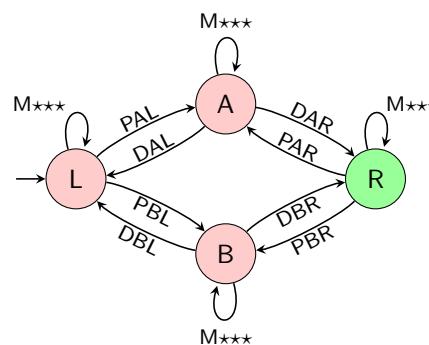
~~ baue feine Abstraktionen aus größeren

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**

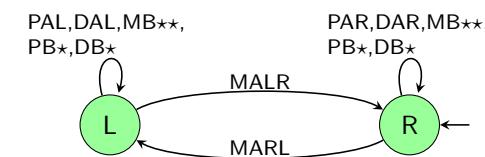
Beispiel: atomare Projektion für das Paket

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



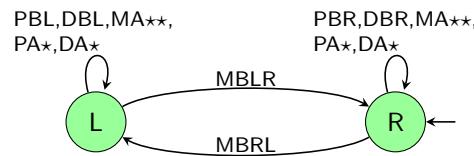
Beispiel: atomare Projektion für Lastwagen A

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



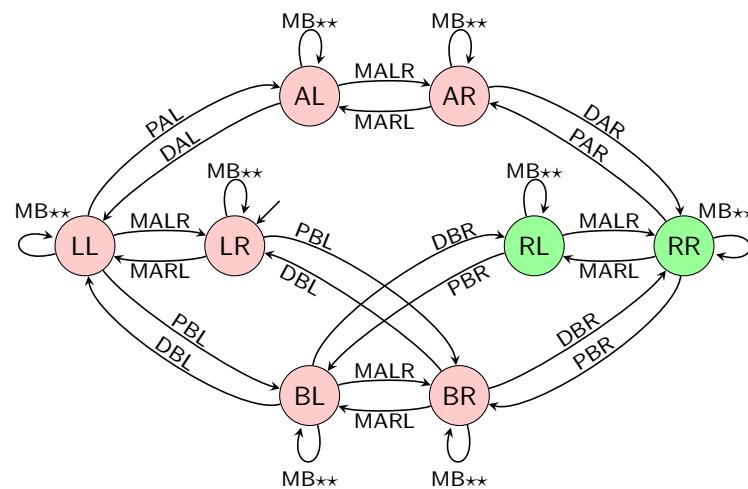
Beispiel: atomare Projektion für Lastwagen B

$\mathcal{S}^{\pi_{\{truck B\}}}$:



Beispiel: synchrones Produkt

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



Synchrones Produkt von Zustandsräumen

Definition (synchrones Produkt von Zustandsräumen)

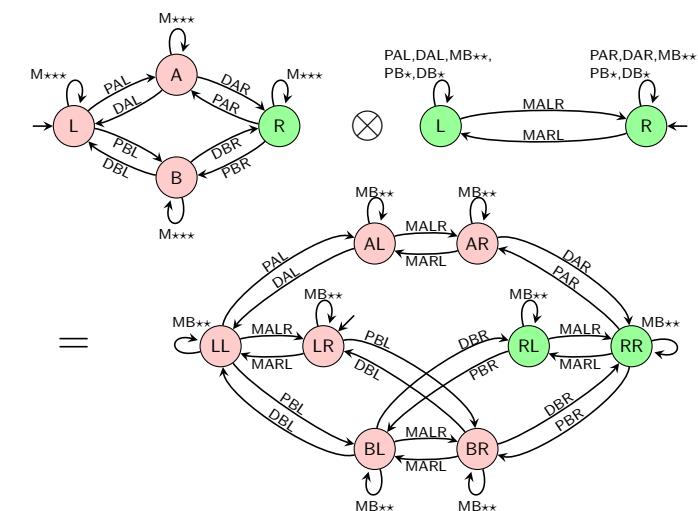
Für $i \in \{1, 2\}$ seien $\mathcal{S}^i = \langle S^i, A, cost, T^i, s_0^i, S_*^i \rangle$ Zustandsräume (mit identischen Aktionen und Kosten).

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

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

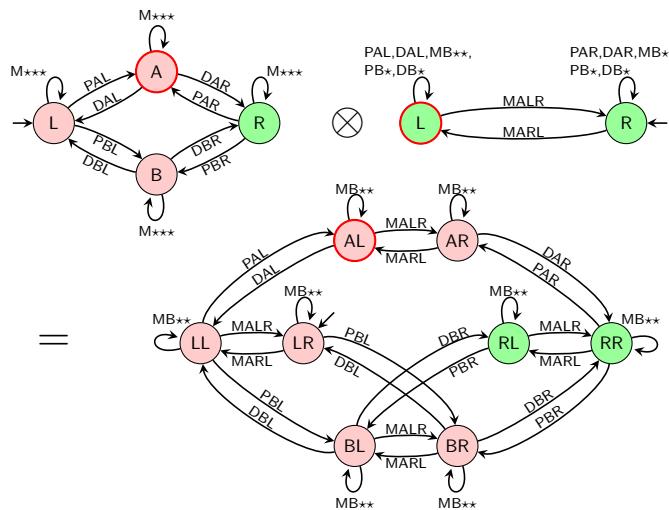
Beispiel: Berechnung des synchronen Produkts

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



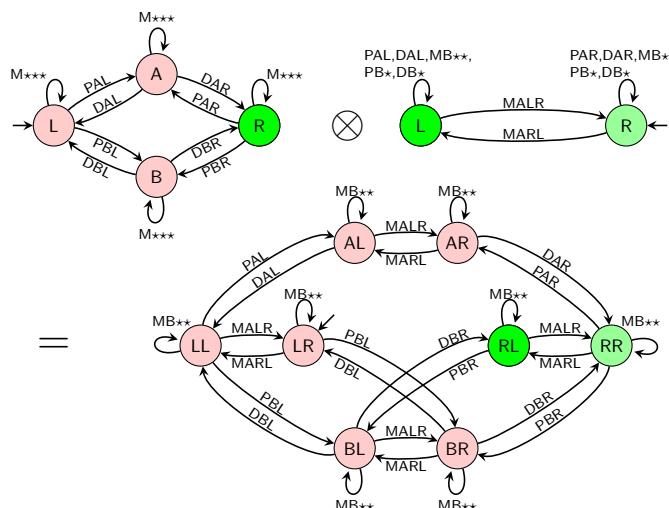
Beispiel: Berechnung des synchronen Produkts

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



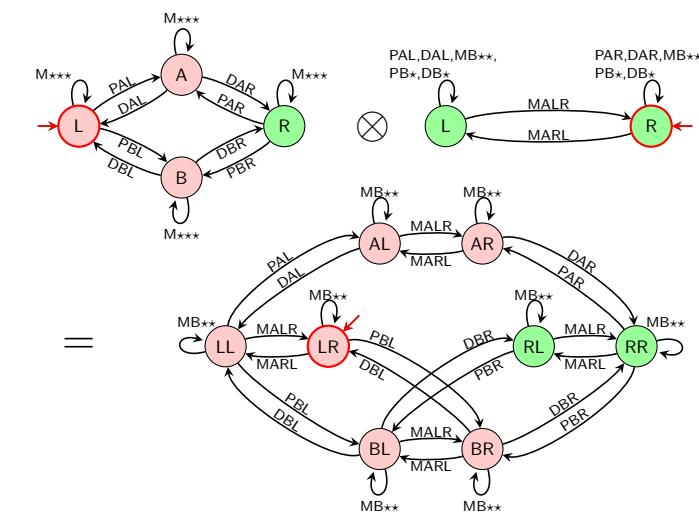
Beispiel: Berechnung des synchronen Produkts

$$\mathcal{S}^{\pi_{\{package\}}} \otimes \mathcal{S}^{\pi_{\{truck A\}}}: S_*^\otimes = S_*^1 \times S_*^2$$



Beispiel: Berechnung des synchronen Produkts

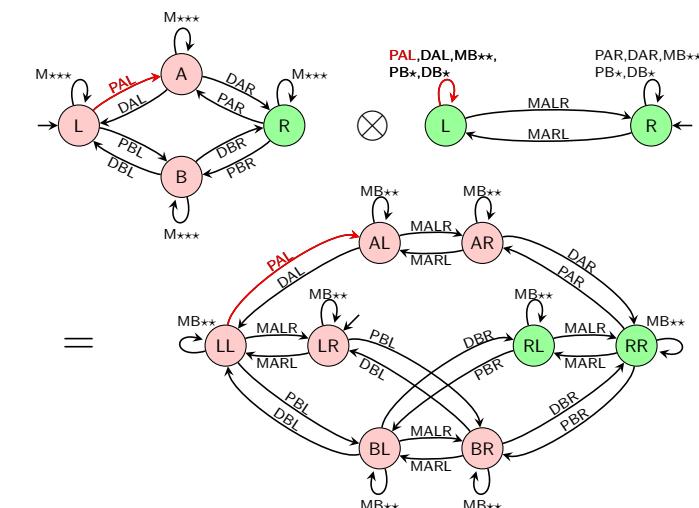
$$\mathcal{S}^{\pi_{\{package\}}} \otimes \mathcal{S}^{\pi_{\{truck A\}}}: s_0^\otimes = \langle s_0^1, s_0^2 \rangle$$



Beispiel: Berechnung des synchronen Produkts

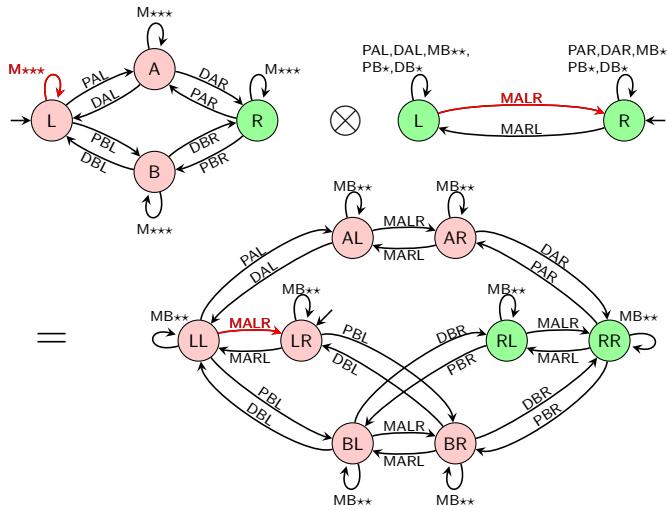
Beispiel: Berechnung des synchronen Produkts

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



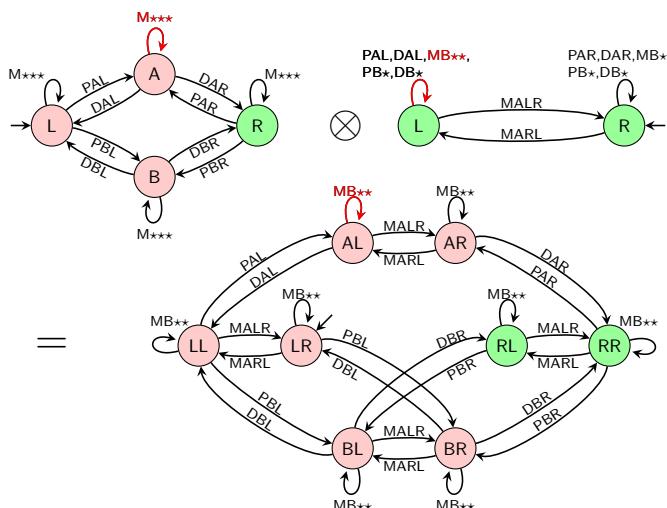
Beispiel: Berechnung des synchronen Produkts

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



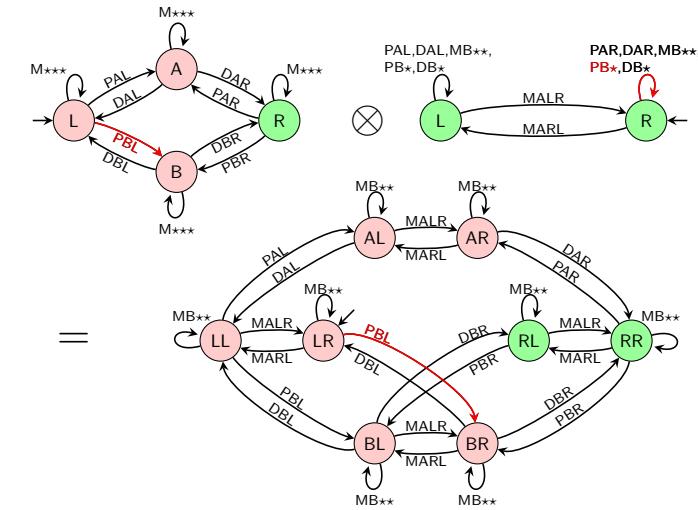
Beispiel: Berechnung des synchronen Produkts

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



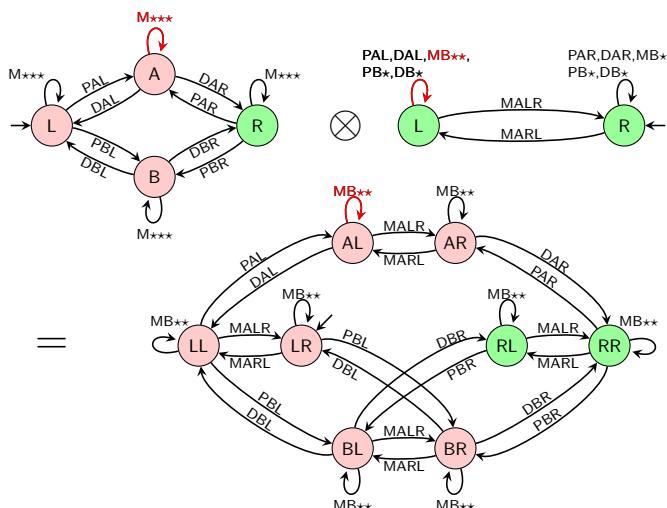
Beispiel: Berechnung des synchronen Produkts

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



Beispiel: Berechnung des synchronen Produkts

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



38.3 Merge-and-Shrink

M&S-Abstraktionen: Kernideen (Fortsetzung)

Kernideen bei M&S:

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

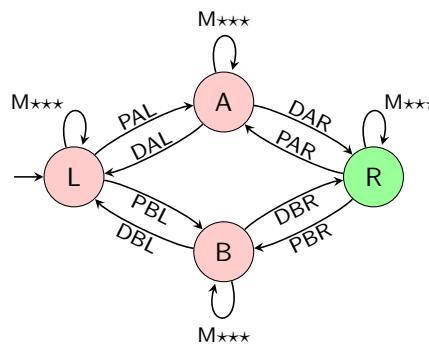
$$\bigotimes_{v \in V} \mathcal{S}^{\pi_{\{v\}}} \text{ ist isomorph zu } \mathcal{S}$$

~~ bau feine Abstraktionen aus gröberen

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

Initialisierungsschritt: atomare Projektion für das Paket

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



Berechnung von M&S-Abstraktionen

Generischer Algorithmus zur Berechnung von M&S-Abstraktionen

```

abs := {Sπ_{\{v\}} | v ∈ V} [Abstraktionen für atomare Projektionen]
while |abs| > 1:
    select S1, S2 from abs
    shrink S1 and/or S2 until size(S1) · size(S2) ≤ K
    abs := abs \ {S1, S2} ∪ {S1 ⊗ S2} [Merge-Schritt]
return the remaining abstraction in abs
  
```

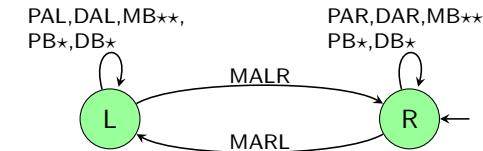
K: Parameter, der max. Anzahl abstrakter Zustände begrenzt

Praktische Implementierungen müssen entscheiden:

- ▶ Welche Abstraktionen werden ausgewählt? ~~ **Merge-Strategie**
- ▶ Wie werden Abstraktionen geschrumpft? ~~ **Shrink-Strategie**
- ▶ Wie soll **K** gewählt werden?

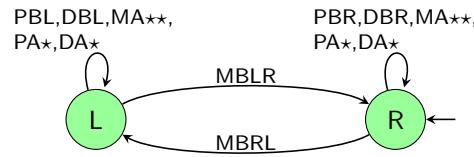
Initialisierungsschritt: atomare Projektion für Lastwagen A

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



Initialisierungsschritt: atomare Projektion für Lastwagen B

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



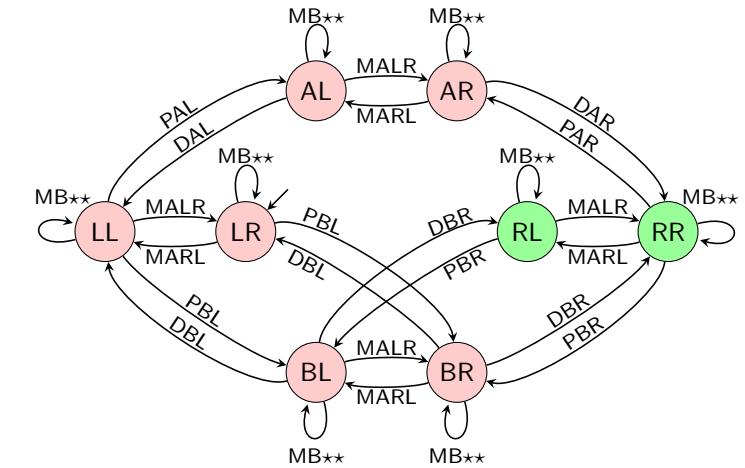
aktuelle Abstraktionsmenge: $abs = \{\mathcal{S}^{\pi\{\text{package}\}}, \mathcal{S}^{\pi\{\text{truck A}\}}, \mathcal{S}^{\pi\{\text{truck B}\}}\}$

Müssen wir vereinfachen?

- Wenn wir genügend Speicher haben, können wir nun $\mathcal{S}^1 \otimes \mathcal{S}^{\pi\{\text{truck B}\}}$ 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 Merge-Schritt

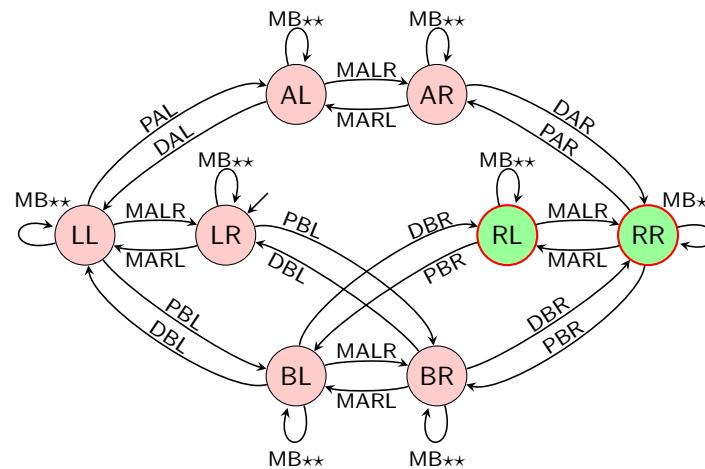
$\mathcal{S}^1 := \mathcal{S}^{\pi\{\text{package}\}} \otimes \mathcal{S}^{\pi\{\text{truck A}\}}$:



aktuelle Abstraktionsmenge: $abs = \{\mathcal{S}^1, \mathcal{S}^{\pi\{\text{truck B}\}}\}$

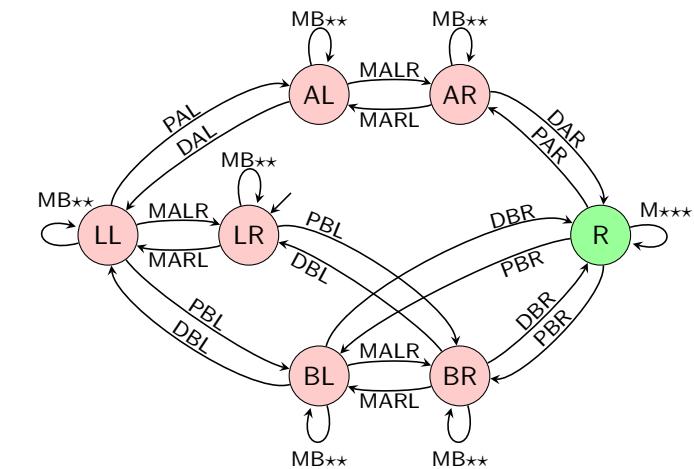
Erster Shrink-Schritt

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



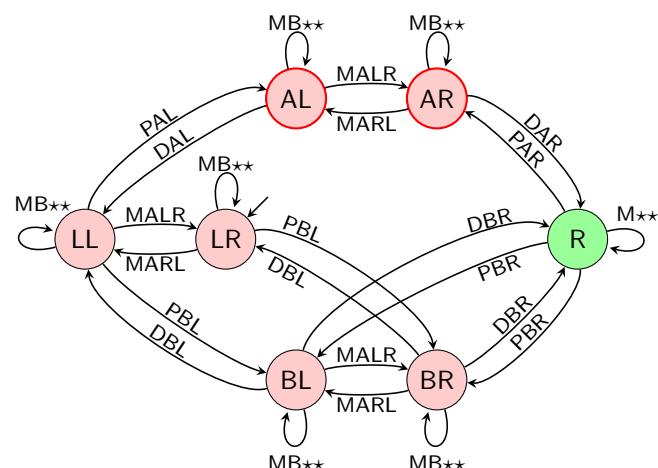
Erster Shrink-Schritt

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



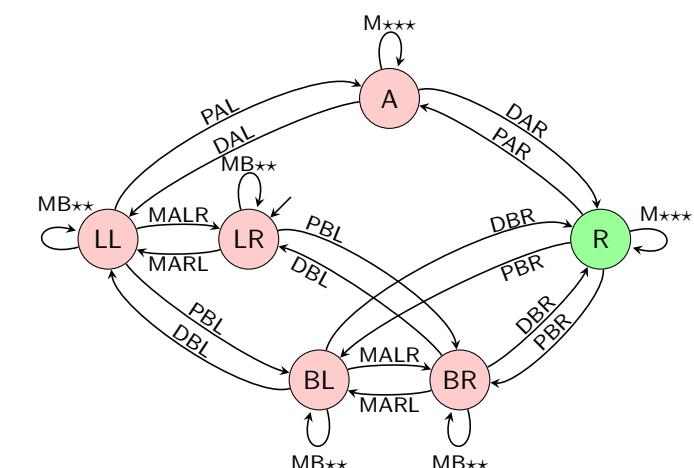
Erster Shrink-Schritt

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



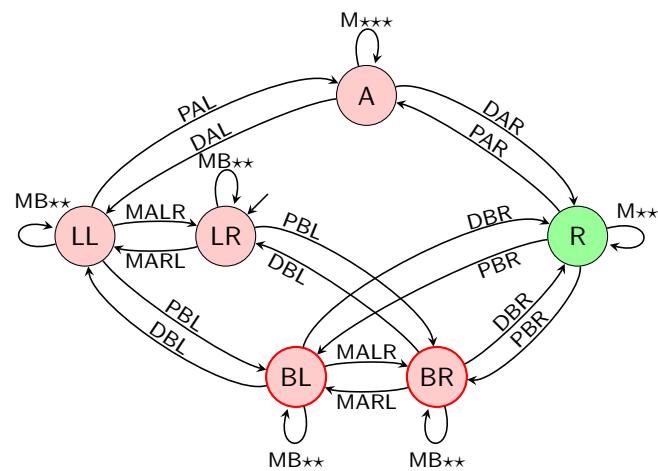
Erster Shrink-Schritt

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



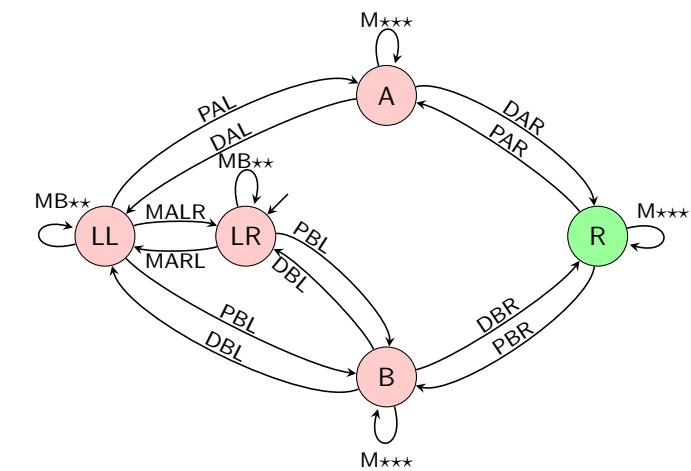
Erster Shrink-Schritt

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



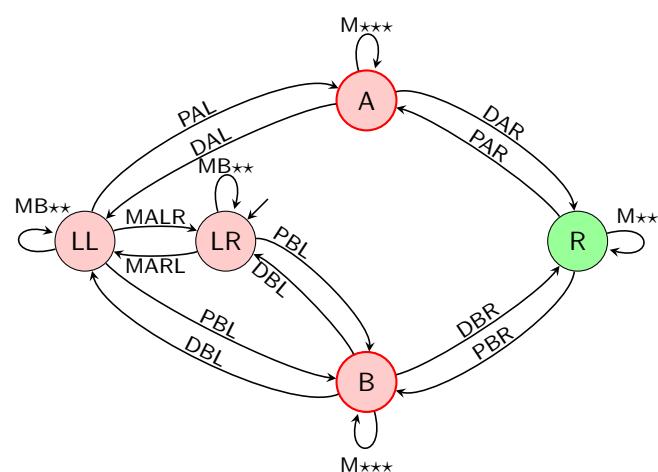
Erster Shrink-Schritt

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



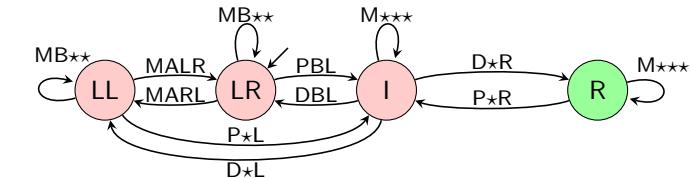
Erster Shrink-Schritt

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



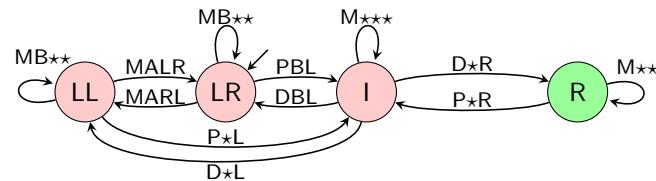
Erster Shrink-Schritt

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



Erster Shrink-Schritt

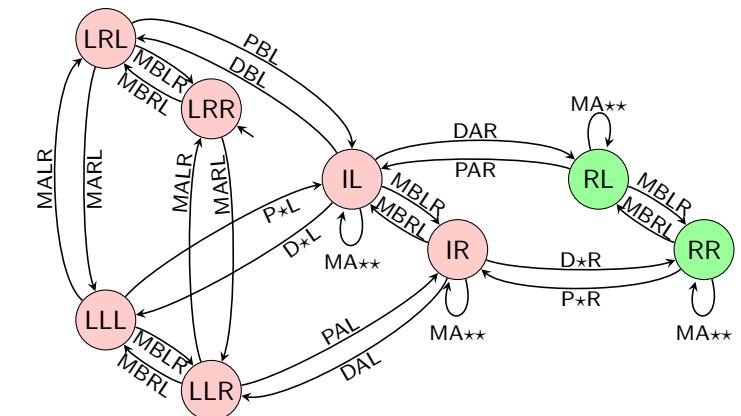
$\mathcal{S}^2 :=$ eine Abstraktion von \mathcal{S}^1



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

Zweiter Merge-Schritt

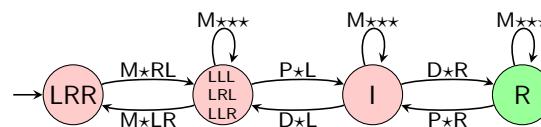
$\mathcal{S}^3 := \mathcal{S}^2 \otimes \mathcal{S}^{\pi\{truck B\}}$:



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

Zweiter Shrink-Schritt

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



- Wir erhalten einen Heuristikwert von 3 für den Anfangszustand \leadsto 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 Grössenschanke 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!

38.4 Zusammenfassung

Zusammenfassung (1)

- ▶ **Merge-and-Shrink-Abstraktionen:** statt wenige Variablen perfekt in der Abstraktion zu berücksichtigen, berücksichtige **alle** Variablen **verlustbehaftet**
- ▶ **synchrones Produkt:** Graphoperation, die zwei abstrakte Transitionssystem zu einem kombiniert
- ▶ Planungsaufgabe lässt sich komplett durch Produkt aus den **atomaren Abstraktionen** rekonstruieren

Zusammenfassung (2)

- ▶ **Merge-and-Shrink:**
 - ▶ ausgehend von allen **atomaren Abstraktionen**
 - ▶ ersetze immer zwei Abstraktionen durch ihr Produkt (**merge**)
 - ▶ verkleinere eine Abstraktion, wenn sie zu gross wird, um in den Speicher zu passen (**shrink**)
- ▶ Praxis: gute **Merge-** und **Shrink-Strategien** wichtig