Grundlagen der Künstlichen Intelligenz

37. Handlungsplanung: Abstraktion und Musterdatenbanken

Malte Helmert

Universität Basel

11. Mai 2015

Planungsheuristiken

Wir besprechen drei Kernideen für allgemeine Heuristiken:

- Delete-Relaxierung
- Abstraktion $\rightsquigarrow$ dieses und nächstes Kapitel
- Landmarken

Grundidee Abstraktion

Schätze Lösungskosten durch Betrachten einer kleineren Planungsaufgabe ab.

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

37.1 SAS$^+$

SAS$^+$-Kodierung

- in diesem und nächstem Kapitel: SAS$^+$-Kodierung statt STRIPS (siehe Kapitel 34)
- Unterschied: Zustandsvariablen nicht alle binär, sondern mit endlichem Wertebereich $\mathrm{dom}(v)$
- 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 $\Pi = \langle V, \mathrm{dom}, I, G, A \rangle$ mit folgenden Komponenten:

- V: endliche Menge von Zustandsvariablen
- dom: Wertebereiche; $\mathrm{dom}(v)$ 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:
  - $\mathit{pre}(a)$: ihre Vorbedingungen (partielle Belegung)
  - $\mathit{eff}(a)$ ihre Effekte (partielle Belegung)
  - $\mathit{cost}(a) \in \mathbb{N}_0$: ihre Kosten

Zustandsraum zu einer SAS$^+$-Planungsaufgabe

Definition (von SAS$^+$-Planungsaufgabe induz. Zustandsraum)

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

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

- 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 $\mathit{pre}(a)$ ist Teilbelegung von s (Vorbedingungen erfüllt)
  - s' entspricht $\mathit{eff}(a)$ 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

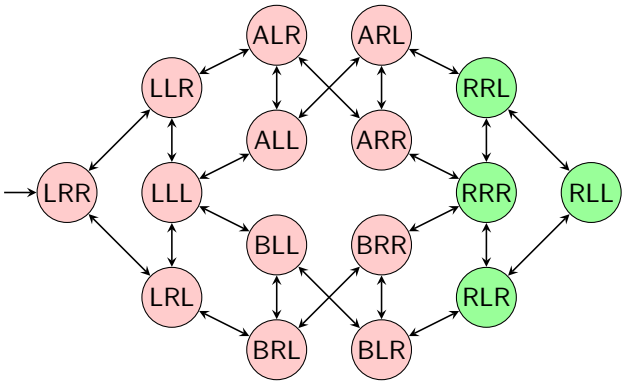
Beispiel (ein Paket, zwei Lastwagen)

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

- $V = \{p, t_\mathrm{A}, t_\mathrm{B}\}$
- $\mathrm{dom}(p) = \{\mathrm{L}, \mathrm{R}, \mathrm{A}, \mathrm{B}\}$ und $\mathrm{dom}(t_\mathrm{A}) = \mathrm{dom}(t_\mathrm{B}) = \{\mathrm{L}, \mathrm{R}\}$
- $I = \{p \mapsto \mathrm{L}, t_\mathrm{A} \mapsto \mathrm{R}, t_\mathrm{B} \mapsto \mathrm{R}\}$ und $G = \{p \mapsto \mathrm{R}\}$
- $A = \{\mathit{pickup}_{i,j} \mid i \in \{\mathrm{A}, \mathrm{B}\}, j \in \{\mathrm{L}, \mathrm{R}\}\}$
  $\cup \{\mathit{drop}_{i,j} \mid i \in \{\mathrm{A}, \mathrm{B}\}, j \in \{\mathrm{L}, \mathrm{R}\}\}$
  $\cup \{\mathit{move}_{i,j,j'} \mid i \in \{\mathrm{A}, \mathrm{B}\}, j, j' \in \{\mathrm{L}, \mathrm{R}\}, j \neq j'\}$ mit:
  - $\mathit{pickup}_{i,j}$ hat Vorbedingungen $\{t_i \mapsto j, p \mapsto j\}$, Effekte $\{p \mapsto i\}$
  - $\mathit{drop}_{i,j}$ hat Vorbedingungen $\{t_i \mapsto j, p \mapsto i\}$, Effekte $\{p \mapsto j\}$
  - $\mathit{move}_{i,j,j'}$ 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 35 (umbenannt, damit Abkürzungen im Folgenden eindeutiger sind)

Zustandsraum für Beispielaufgabe



- Zustand $\{p \mapsto i, t_\mathrm{A} \mapsto j, t_\mathrm{B} \mapsto k\}$ abgebildet als ijk.
- Kantenbeschriftungen der Übersicht halber weggelassen.
  Zum Beispiel hat die Kante von LLL zu ALL die Beschriftung $\mathit{pickup}_{\mathrm{A,L}}$.

37.2 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 $\mathcal{S}^\alpha$, der ähnlich zu $\mathcal{S}$ ist, aber kleiner.

Abstraktionsheuristik

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

Induzierte Abstraktion

Definition (induzierte Abstraktion)

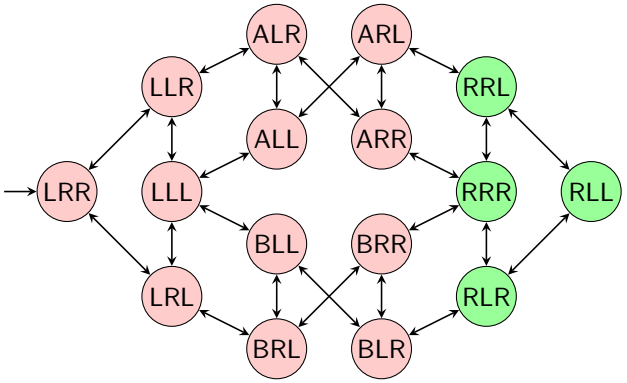
Sei $\mathcal{S} = \langle S, A, cost, T, s_0, S_\star \rangle$ ein Zustandsraum,
und sei $\alpha : S \to S'$ eine surjektive Funktion.

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:

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

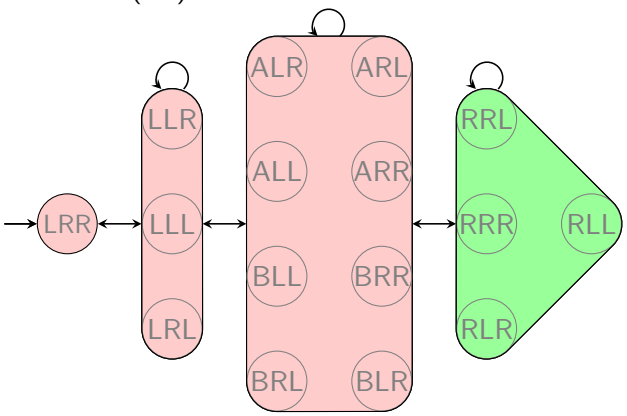
Abstraktion: Beispiel

konkreter Zustandsraum



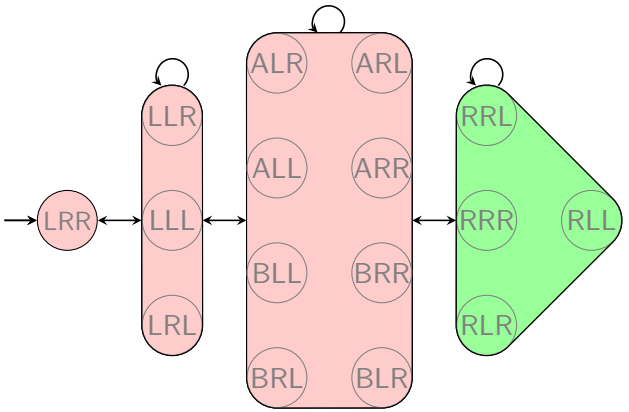
Abstraktion: Beispiel

(ein) abstrakter Zustandsraum



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

Abstraktionsheuristik: Beispiel

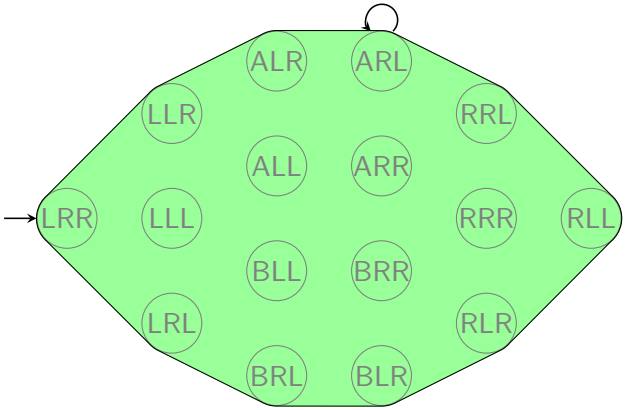


$h^\alpha(\{p \mapsto \mathrm{L}, t_\mathrm{A} \mapsto \mathrm{R}, t_\mathrm{B} \mapsto \mathrm{R}\}) = 3$

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?

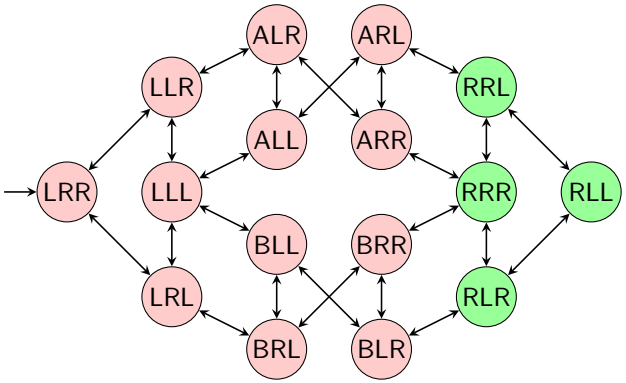
Meist schlechte Idee: Ein-Zustand-Abstraktion



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)

37.3 Musterdatenbanken

Musterdatenbanken: Hintergrund

- Die am häufigsten verwendeten Abstraktionsheuristiken sind Musterdatenbank-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: Projektionen

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.

Formalisiert durch Projektionen. Beispiel:

- $s = \{v_1 \mapsto d_1, v_2 \mapsto d_2, v_3 \mapsto d_3\}$
- Projektion auf $P = \{v_1\}$ (= ignoriere v_2, v_3):
  $\alpha(s) = s|_P = \{v_1 \mapsto d_1\}$
- Projektion auf $P = \{v_1, v_3\}$ (= ignoriere v_2):
  $\alpha(s) = s|_P = \{v_1 \mapsto d_1, v_3 \mapsto d_3\}$

Musterdatenbanken: Definition

Definition (Musterdatenbank-Heuristik)

Sei P eine Teilmenge der Variablen einer Planungsaufgabe.

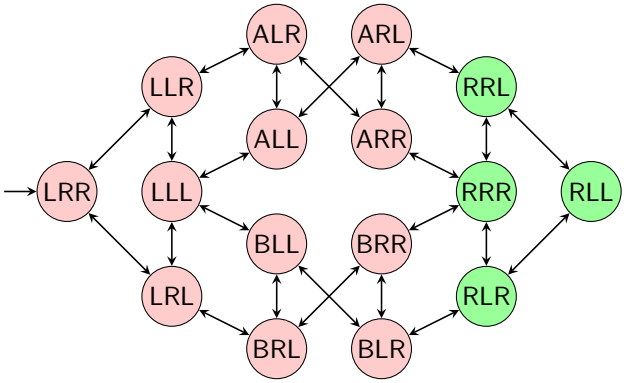
Die von der Projektion π_P auf P induzierte Abstraktionsheuristik heisst Musterdatenbank-Heuristik (PDB-Heuristik) mit Muster P.

Kurzschreibweise: h^P für h^{π_P}

Anmerkungen:

- „PDB“ von englisch pattern database; „Muster“ = pattern
- „Musterdatenbank“ in Analogie zu Endspieldatenbanken (erfolgreich eingesetzt für 2-Personen-Spiele)

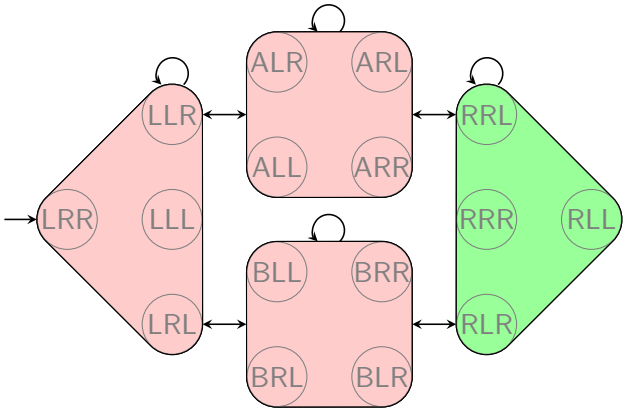
Beispiel: konkreter Zustandsraum



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

Beispiel: Projektion (1)

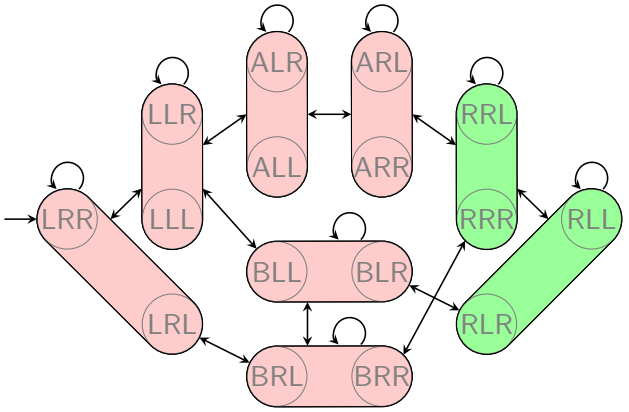
Von $\pi_{\{package\}}$ induzierte Abstraktion:



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

Beispiel: Projektion (2)

Von $\pi_{\{package,truck\ A\}}$ 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
    $\rightsquigarrow$ Platzaufwand = Anzahl abstrakter Zustände

37.4 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.