

# Grundlagen der Künstlichen Intelligenz

## 33. Handlungsplanung: Einführung

Malte Helmert

Universität Basel

4. Mai 2015

# Grundlagen der Künstlichen Intelligenz

4. Mai 2015 — 33. Handlungsplanung: Einführung

## 33.1 Einführung

## 33.2 Wiederholung: Zustandsräume

## 33.3 Kompakte Beschreibungen

## 33.4 Zusammenfassung

## Einordnung

Einordnung:

Handlungsplanung

Umgebung:

- ▶ **statisch** vs. dynamisch
- ▶ **deterministisch** vs. nicht-deterministisch vs. stochastisch
- ▶ **vollständig** vs. partiell vs. nicht **beobachtbar**
- ▶ **diskret** vs. stetig
- ▶ **ein Agent** vs. mehrere Agenten

Lösungsansatz:

- ▶ problemspezifisch vs. **allgemein** vs. lernend

## 33.1 Einführung

## Handlungsplanung

### Was ist Handlungsplanung?

„Planning is the art and practice of thinking before acting.“

— P. Haslum

↔ Finden von **Plänen** (Aktionsfolgen), die von einem Anfangszustand aus einen Zielzustand erreichen

Unser Thema ist **klassische Handlungsplanung**:

- ▶ **allgemeiner** Ansatz für Lösung von „Suchproblemen“ (aus Kapiteln 5–20)
- ▶ **klassisch** = statisch, deterministisch, vollständig beobachtbar
- ▶ **Varianten**: probabilistisches Planen, Planen mit partieller Beobachtbarkeit, Online-Planen, ...

## Handlungsplanung informell

### Gegeben:

- ▶ Beschreibung eines Zustandsraums in einer geeigneten Beschreibungssprache (**Planungsformalismus**)

### Gesucht:

- ▶ ein **Plan**, d. h. Lösung für den beschriebenen Zustandsraum (Folge von Aktionen vom Anfangszustand zum Ziel)
- ▶ oder ein Nachweis, dass kein Plan existiert

### Unterscheidung zwischen

- ▶ **optimalen Planern**: garantieren, dass gefundene Pläne optimal sind, d. h. minimale Gesamtkosten haben
- ▶ **suboptimalen Planern (satisficing)**: dürfen suboptimale Pläne liefern

## Was ist neu?

Viele Probleme, die wir schon kennen, sind im Prinzip Planungsprobleme:

- ▶ Blocks world
- ▶ Missionare und Kannibalen
- ▶ 15-Puzzle

Neu ist, dass wir uns nun für **allgemeine** Algorithmen interessieren, d. h. der Entwickler des Suchalgorithmus **kennt nicht** die zu lösenden Probleme.

- ↔ keine problemspezifischen Heuristiken!
- ↔ **Eingabesprache**, die zu lösende Probleme modelliert

## Handlungsplanung: Überblick

### Kapitelüberblick:

- ▶ **33. Einführung**
- ▶ 34. Planungsformalisten
- ▶ 35.–40. Planungsheuristiken

## 33.2 Wiederholung: Zustandsräume

## Über diesen Abschnitt

Nichts neues hier!

Dieser Abschnitt ist eine **Wiederholung** von Abschnitt 5.2 aus dem Kapitel „Klassische Suche: Zustandsräume“.

## Formalisierung von Zustandsräumen

Vorbemerkungen:

- ▶ Um Suchprobleme sauber algorithmisch fassen zu können, benötigen wir eine **formale Definition**.
- ▶ grundlegendes semantisches Konzept: **Zustandsräume**
- ▶ Zustandsräume sind (annotierte) **Graphen**
- ▶ **Pfade** zu Zielzuständen entsprechen **Lösungen**
- ▶ **kürzeste Pfade** entsprechen **optimalen Lösungen**

## Zustandsräume

Definition (Zustandsraum)

Ein **Zustandsraum** ist ein 6-Tupel  $\mathcal{S} = \langle S, A, cost, T, s_0, S_* \rangle$  mit

- ▶  $S$  endliche Menge von **Zuständen**
- ▶  $A$  endliche Menge von **Aktionen**
- ▶  $cost : A \rightarrow \mathbb{R}_0^+$  **Aktionskosten**
- ▶  $T \subseteq S \times A \times S$  **Transitionsrelation** oder Übergangsrelation; **deterministisch in  $\langle s, a \rangle$**  (siehe nächste Folie)
- ▶  $s_0 \in S$  **Anfangszustand**
- ▶  $S_* \subseteq S$  Menge der **Zielzustände**
- ▶ **auch**: Transitionssystem (transition system)
- ▶ **englisch**: state space, state, action, action costs, transition relation, initial state, goal states

## Zustandsräume: Transitionen, Determinismus

### Definition (Transition, deterministisch)

Sei  $\mathcal{S} = \langle S, A, cost, T, s_0, S_* \rangle$  ein Zustandsraum.

Die Tripel  $\langle s, a, s' \rangle \in T$  heissen **Transitionen/Zustandsübergänge**.

Wir sagen  $\mathcal{S}$  **hat die Transition**  $\langle s, a, s' \rangle$ , falls  $\langle s, a, s' \rangle \in T$  und schreiben dafür  $s \xrightarrow{a} s'$  sowie  $s \rightarrow s'$ , wenn  $a$  nicht interessiert.

Transitionen sind **deterministisch** in  $\langle s, a \rangle$ :  $s \xrightarrow{a} s_1$  und  $s \xrightarrow{a} s_2$  mit  $s_1 \neq s_2$  ist nicht erlaubt.

## Zustandsräume: Begriffe

### Begriffe:

- ▶ Vorgänger, Nachfolger
- ▶ anwendbare Aktionen
- ▶ Pfad, Länge, Kosten
- ▶ erreichbar
- ▶ Lösung, optimale Lösung

## 33.3 Kompakte Beschreibungen

## Zustandsräume mit deklarativen Repräsentationen

Wie kommt der Zustandsraum in den Computer?

**vorher:** als Black Box

**jetzt:** als deklarative Beschreibung

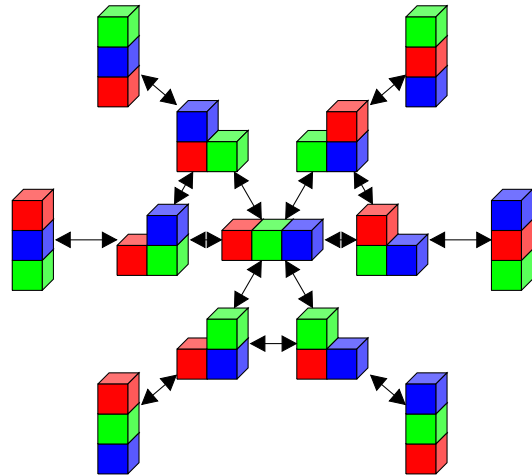
Erinnerung: Kapitel 6

Zustandsräume mit deklarativen Repräsentationen

Stelle Zustandsräume **deklarativ** dar:

- ▶ **kompakte** Beschreibung des Zustandsraums als Eingabe für Algorithmen  
 $\rightsquigarrow$  Zustandsraum **exponentiell grösser** als Eingabe
- ▶ Algorithmen arbeiten **direkt auf kompakter Beschreibung**
- $\rightsquigarrow$  erlaubt automatische Problemumformulierung, Vereinfachung, Abstraktion, usw.

## Erinnerung: Blocks world



Problem:  $n$  Blöcke  $\rightsquigarrow$  mehr als  $n!$  Zustände

## Kompakte Beschreibung von Zustandsräumen

Wie können wir Zustandsräume kompakt beschreiben?

Kompakte Beschreibung von vielen Zuständen

- ▶ Führe **Zustandsvariablen** ein.
- ▶ Zustände sind Zuweisungen an die Zustandsvariablen.
- $\rightsquigarrow$  z. B.  $n$  binäre Zustandsvariablen können  $2^n$  Zustände beschreiben
- ▶ **Transitionen** und **Ziel** werden über Logik oder logikähnliche Formalismen kompakt beschrieben.

verschiedene Varianten: verschiedene **Planungsformalismen**

## 33.4 Zusammenfassung

## Zusammenfassung

- ▶ **Handlungsplanung**: Suche in **allgemeinen** Zustandsräumen
- ▶ **Eingabe**: kompakte, deklarative Beschreibung des Zustandsraums