

# PLANEN

Planen: Strips und Nachfolger

# Einführung

## Grundidee

- Was ist ein Plan ?
  - ▶ Zeitabhängig
  - ▶ Startzustand  $\longrightarrow$  Zielzustand
  - ▶ Folge von Aktionen zum Erreichen
  - ▶ Auffinden der richtigen Aktionen und Reihenfolge durch einen Computer

# Einführung

## Grundlage - Situationskalkül

- Vorher: Beschreibung der „Welt“ durch Prädikatenlogische Ausdrücke
- Problem, dass nur aus diesen Ausdrücken kein Zielzustand abgeleitet werden kann

# Einführung

## Grundlage - Situationskalkül

- Einführung relationaler Fluents
- Ausdrücke beziehen sich auf eine Situation  
 $\text{ontable}(x, y, S)$
- Relationale Fluents beschreiben die Welt

# Einführung

## Grundlage - Situationskalkül

- Aktionen sind funktionale Ausdrücke
- Aktionen werden angewendet durch :  
 $\text{do}(\text{Aktion}, \text{Situation})$
- Es wird eine neue Situation erzeugt

# Einführung

## Grundlage - Situationskalkül

- Jede Aktion erhält Ausführungsbedingungen
- Für jeden Relationalen Fluent muss der Effekt einer Aktion definiert werden
- Effektaxiome
- Kann zu vielen Definitionen führen

# Einführung

## Grundlage - Situationskalkül

- Finden eines Planes mittels Resolution

# Einführung

## Grundlage - Situationskalkül

- Probleme !
- Rahmenproblem
- Qualifikationsproblem
- Verzweigungsproblem

# Einführung

## Situationskalkül - Probleme

- Rahmenproblem :
  - ▶ Gültigkeit von Fluenten auf die nachfolgende Situation übertragen
  - ▶ Für jedes Operator - Aktions - Paar anzugeben
  - ▶ Stellt die zentrale Schwierigkeit beim Planen dar
  - ▶ Jeder Planungsalgorithmus muss eine effiziente Lösung dieses Problems bereitstellen !

# Einführung

## Situationskalkül - Probleme

- Qualifikationsproblem :
  - ▶ Alle Bedingungen eines Zustands der realen Welt sollten angegeben werden
  - ▶ Übertragen der Planergebnisse auf die reale Welt
  - ▶ Lösungsansätze mit nichtmonotonem Schließen, also formulieren von Standardannahmen
  - ▶ Beispiel: Vögel fliegen normalerweise

# Einführung

## Situationskalkül - Probleme

- Verzweigungsproblem :
  - ▶ Indirekte Effekte von Aktionen erkennen
  - ▶ Lösung mittels Interferenzverfahren

# STRIPS

## Geschichte

- STanford Research Institute Problem Solver
- Entwickelt: Richard Fikes & Nils Nilsson
- Klassisches Planen

# S T R I P S

Idee

- Vermeidung der Probleme des Situationskalküls
- Beibehalten der Zustandsrepräsentation

# STRIPS

## Zustand - Zielzustand

- Zustände werden durch Mengen von Prädikaten dargestellt
- Wird STRIPS-Datenbasis genannt
- Zielzustand ist eine STRIPS-Datenbasis
- Beispiel: { raum\_leer(A), kiste\_in(a,B), beweglich(a) }

# STRIPS

## Operatoren

- Operatoren verändern direkt die Datenbasis
- Operatoren haben Namen
- Besteht aus 3 Teilen:
  - i) Menge C: Vorbedingungen
  - ii) Menge D: Entfernungsliste
  - iii) Menge A: Hinzufügeliste

# STRIPS

## Operatoren

- Beispiel:

schiebe\_in\_raum(x, y, z)

C: raum\_leer(y), beweglich(x)

D: raum\_leer(y), kiste\_in(z)

A: kiste\_in(y)

# STRIPS

## Operatoren

- Alle Literale aus der aktuellen Datenbasis bleiben erhalten, wenn sie nicht in  $D$  sind
- Neuer Zustand ergibt sich:  $S_2 = Op(S_1)$

# STRIPS

Ein Plan

- Ein Plan ergibt sich aus einer Operatorliste:
  - ➔  $P = [Op1, Op2, \dots, Opn]$
- Pläne können Konkateniert werden, also Hintereinanderschalten
- Plan ergibt sich als Lösung eines Suchproblems

# STRIPS Algorithmus

- Zwei Möglichkeiten:
  - ▶ Vorwärtssuche
  - ▶ Rückwärtssuche

# STRIPS

## Algorithmus - Vorwärtssuche

- Solange Operatoren anwenden bis Zieldatenbasis erzeugt wurde
- Nicht praktikabel
- Ziel wird nicht berücksichtigt

# STRIPS

## Algorithmus - Rückwärtssuche

- Rückwärtssuche arbeitet zielgerichtet
- Es wird ein Teilziel gewählt und nachfolgend ein entsprechender Operator, der dieses Ziel erfüllt
- Vorbedingungen des Operators müssen geprüft werden

# STRIPS

## Algorithmus - Rückwärtssuche

### Eingabe:

Z Ziel ( Menge von Grundliteralen )  
Start Datenbasis

### Ausgabe:

P Plan

### R-Strips

P := [ ]

S := Start

while ( Z  $\not\subset$  S )

g := Element aus Z das nicht in S vorkommt

Op := Ein Operator das gilt: g  $\in$  A

Pc := R-Strips ( {Op}, S )

S := Op( Pc( S ) )

P := P + Pc + [ Op ]

endwhile

return P

# Aktuellere Planungsansätze

- Viele verschiedene Ansätze
- ABSTRIPS
- GraphPlan
- Nichtlineares Planen
- Agenten

# Aktuellere Planungsansätze

- **ABSTRIPS:**
  - ▶ Erstes System, das automatisch eine hierarchische Ordnung erstellt
  - ▶ Von abstrakten Plänen hin zu detaillierteren Plänen
  - ▶ Weiterentwicklungen
  - ▶ Hierarchisches Planen

# Aktuellere Planungsansätze

- GraphPlan
  - ▶ Graphen werden zum Planen benutzt
  - ▶ Zwei Phasen: Expandieren & Extrahieren
  - ▶ Schnell dadurch eine Sensation gewesen
  - ▶ Viele Weiterentwicklungen die noch schneller sind

# Aktuellere Planungsansätze

- Nichtlineares Planen:
  - ▶ Suchraum: Unvollständige Pläne
  - ▶ Durch Plantransformation werden Teilpläne zu erweiterten Plänen

# Aktuellere Planungsansätze

- Agentensysteme:
  - ▶ RHINO - Herumführen von Museumsbesuchern

# Fragen ?



# Quellen

- Christoph Beierle, Gabriele Kern-Isberner: Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen, Kapitel 10.
- GraphPlan: <http://www.cs.cmu.edu/~avrim/graphplan.html>
- ABSTRIPS: <http://www.isi.edu/integration/papers/knoblock92-abstrips.pdf>
- ABSTRIPS: <http://www.dtic.mil/cgi-bin/GetTRDoc?D=ADA269528&Location=U2&doc=GetTRDoc.pdf>