# Planen als Erfüllbarkeitsproblem

Martin Hofmann

19.01.2007

Seminar Planen und Problemlösen

# Übersicht

# Zustandsraum vs. Axiome

## Zustandsraumbasiert (↻ STRIPS):

- Planungsdomäne: Menge von Zuständen und Übergängen
- Plan: gültigen Aktionsfolge von $s_0$ nach $s_n$
- Suchraum: Zustandsraum

## Axiombasiert

- logische Formeln beschreiben Planungsproblem (FOL)
- Existenz eines Planes von $s_0$ nach $s_n$ wird bewiesen
- Aktionsfolge wird konstruktiv erstellt (quasi Beiprodukt)

# Situation Calculus Axiomatisierung

- Green (1964), McCarthy (1983)
- Aussagen als Prädikate (*fluents*)
- Aktionen als Implikationen
- Zustandsvariable $S$ speichert Teilpläne
- Planen durch Resolution in FOL

**Beispiel**

$\forall A, B, S. frei(A, S) \land frei(B, S) \rightarrow auf(A, B, hebe(A, B, S))$

# Planungsproblem

### Start

$$frei(b_1, s_0) \qquad \text{(S1)}$$
$$frei(b_2, s_0) \qquad \text{(S2)}$$

### Ziel

$$auf(b_1, b_2, S) \qquad \text{(Z)}$$

### $A_{hebe}$

$$\neg frei(A, S) \vee \neg frei(B, S) \vee auf(A, B, hebe(A, B, S)) \qquad \text{(A)}$$

$\rightarrow$ Beweis durch Widerspruch: $\neg auf(b_1, b_2, S) \vee Plan(S)$

# Resolution

1. $frei(b_1, s_0)$ (S1)
2. $\neg frei(A, S) \lor \neg frei(B, S) \lor auf(A, B, hebe(A, B, S))$ (A)
3. $\neg frei(B, s_0) \lor auf(A, B, hebe(b_1, B, s_0))$ (Res. $1 + 2$)
4. $frei(b_2, s_0)$ (S2)
5. $auf(A, B, hebe(b_1, b_2, s_0))$ (Res. $3 + 4$)
6. $\neg auf(b_1, b_2, S) \lor Plan(S)$ (Behauptung)
7. $Plan(hebe(b_1, b_2, s_0))$ Widerspruch, Res. $5 + 6$

# Grenzen des Situation Calculus

- Deduktives Planen in FOL
  - $hebe(A, B, hebe(C, D, hebe(E, F, (\dots))))$
  - Zustandsterme unendlicher Länge
  - Suchraum unbeschränkt
- Ineffizient, skaliert sehr schlecht
- kaum anwendbar auf praktische Probleme
- zustandsbasierte Planer schneller, effektiver, besser
- Paradigmenwechsel $\rightarrow$ STRIPS

*“Planen nur mit spezialisierten Systemen”*

# Deduktiv vs Erfüllbarkeitsbasiert

- Deduktives Planen
  - $\longrightarrow$ Plan folgt deduktiv aus Axiomen, Start- und Endzustand
- Erfüllarkeitsbasiertes Planen (SAT-Planen)
  - $\longrightarrow$ gültiger Plan ist Modell aller Axiome
- Modell ist Variablenbelegung, die eine Formel erfüllt

**Beispiel**

F: $A \rightarrow B$

M: $A \hookleftarrow \top, B \hookleftarrow \top$
$A \hookleftarrow \bot, B \hookleftarrow \top$
$A \hookleftarrow \bot, B \hookleftarrow \bot$

# SAT Planen nach Kautz & Selman

✓

- Modell für deduktive Axiome suchen!

✗

- FOL ist unentscheidbar
- Kein vollständiger Algorithmus existiert!

✓

- Verwende Aussagenlogik!

# FOL Axiome in Aussagenlogik

- Fixierung auf diskreten Zeitpunkt $t$
- $t \leq$ maximaler Planlänge $N$
- fluents und Aktionen sind Aussagen $\in \{\top, \bot\}$
- *keine* Terme oder Funktionen

### *fluents*

$$auf(b_1, b_2, 3) \qquad frei(b_1, 0)$$

### *Aktionen*

$$\forall A, B, I. frei(A, I) \wedge frei(B, I) \wedge hebe(A, B, I) \rightarrow auf(A, B, I+1)$$

# SAT Planen nach Kautz & Selman

✓

- Verwende Aussagenlogik!

✗

- SAT ist NP-hard und PSPACE-complete!

# SAT – NP-hard

- Axiome expandieren zur Konjunktion aller Instantiierungen
- Zahl der Variablen $|Vars| = n|Ops||Dom|^{A_o}$
- Rechenzeit = $O(k^{|Vars|})$

$\forall A, B, I. frei(A, I) \wedge frei(B, I) \wedge hebe(A, B, I) \rightarrow auf(A, B, I + 1)$

$$\begin{gathered}
frei(b_1, 0) \wedge frei(b_1, 0) \wedge hebe(b_1, b_1, 0) \rightarrow auf(b_1, b_1, 1) \\
frei(b_1, 0) \wedge frei(b_2, 0) \wedge hebe(b_1, b_2, 0) \rightarrow auf(b_1, b_2, 1) \\
frei(b_2, 0) \wedge frei(b_1, 0) \wedge hebe(b_2, b_1, 0) \rightarrow auf(b_2, b_1, 1) \\
\ldots \\
frei(b_k, N) \wedge frei(b_k, N) \wedge hebe(b_k, b_k, N) \rightarrow auf(b_k, b_k, N)
\end{gathered}$$

# SAT Planen nach Kautz & Selman

✓

- Für die Praxis irrelevant
- bis zur Planlänge *n* suchen
- effizient Kodieren

✗

- Woher weiß man die Planlänge?

✓

- *iterative deepening*!

# SAT Planen nach Kautz & Selman

✓

- *iterative deepening*!

✖

- Unerwünschte Modelle!

# Unerwünschte Modelle

- alle Axiome sind in Blockwelt war
- nicht alle Welten in denen sie gelten entsprechen einer Blockwelt

  $\longrightarrow$ nur deduktiv benutzbar

$\forall A, B, I. frei(A, I) \wedge frei(B, I) \wedge hebe(A, B, I) \rightarrow auf(A, B, I+1)$

- $frei(b_1, 0), frei(b_2, 0), auf(b_1, b_2, 1)$

  $\longrightarrow$ Effekt ohne Aktion
- $frei(b_1, 0), hebe(b_1, b_2, 0), auf(b_1, b_2, 1)$

  $\longrightarrow$ Aktion trotz verletzter Prämisse

# SAT Planen nach Kautz & Selman

✔
- *iterative deepening*!

✖
- Unerwünschte Modelle!

✔
- Axiome anpassen!

# Axiomatisierung (1)

## Aktionen

$$\forall A, B, I. frei(A, I) \wedge frei(B, I) \wedge auf(A, C, I) \wedge hebe(A, B, I) \rightarrow \\ auf(A, B, I+1) \wedge frei(C, I+1)$$

## Aktionen implizieren ihre Vorbedingungen und Effekte

$$\forall A, B, I. hebe(A, B, I) \rightarrow \\ frei(A, I) \wedge frei(B, I) \wedge auf(A, C, I) \wedge \\ auf(A, B, I+1) \wedge frei(C, I+1)$$

## Wechselseitiger Ausschluss

$$\forall A, A', B, B', I. (A \neq A' \vee B \neq B') \rightarrow \\ \neg hebe(A, B, I) \vee \neg hebe(A', B', I)$$

# Axiomatisierung (2)

## mindestens eine Aktion

$$\forall I < n. \exists A, B. hebe(A, B, I)$$

## Startzustand vollständig (explizite CWA)

$$\begin{gathered} frei(b_1, 1) \land \neg frei(b_2, 1) \land frei(tisch, 1) \\ \land auf(b_1, b_2, 1) \land \neg auf(b_2, b_1, 1) \land \\ \neg\neg auf(b_1, b_1, 1) \land \neg auf(b_2, b_2, 1) \land \\ \neg auf(b_1, tisch, 1) \land auf(b_2, tisch, 1) \end{gathered}$$

## Zielzustand partiell

$$auf(b_2, b_1, 3)$$

# SAT-Plan Techniken

- Expansion jeder Formel
  - $\forall$: Konjunktion aller Instanzen
  - $\exists$: Disjunktion aller Instanzen
- Modell
  - ist gültige Variablenbelegung
  - repräsentiert vollständig geordnete Sequenz von Aktionen
- Klauseln in KNF

$\rightarrow$ jede einzelne ist zu erfüllen

# Systematische Methoden – Davis-Putnam

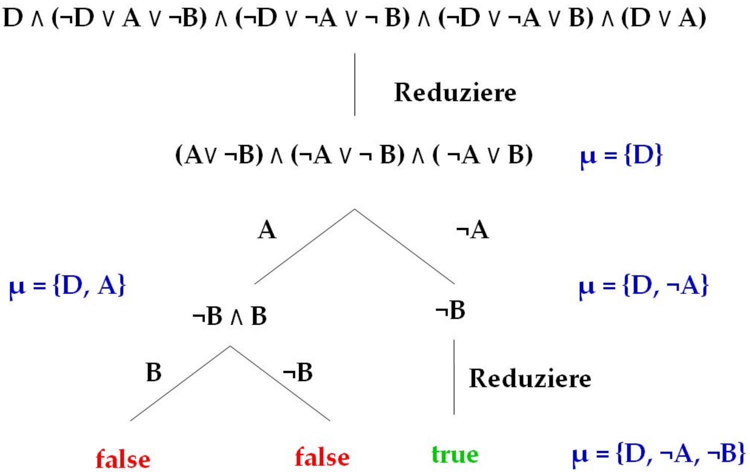
*Systematische Methoden sind vollständig und korrekt*

## Davis-Putnam-Algorithmus

Generiere alle Resolventen und teste ob $\emptyset$ erzeugt wurde

1. solange Einheitsklauseln und Einheitsliterale $E$ vorhanden
   - $E := \top$
   - lösche $\neg E$ aus allen Klauseln
   - lösche alle Klauseln $E \in K$
2. wähle Variable $V$
   - $K \cup \{V\}$ und $K \cup \{\neg V\}$
   - goto 1.
   - backtracking bei Widerspruch

# Davis-Putnam – Beispiel (allgemein)



# Davis-Putnam – Beispiel (Blockwelt)

## Klauseln

$$hebe(b_2, b_1, 1) \rightarrow$$
$$frei(b_2, 1) \wedge frei(b_1, 1) \wedge auf(b_2, tisch, 1) \wedge$$
$$auf(b_2, b_1, 2) \wedge frei(tisch, 2)$$

1. in $s_0$ gelte $\neg frei(b_2, 1)$
2. Klausel wird zu $\neg hebe(b_2, b_1, 1)$ reduziert (Schritt 1. in DP)
3. hebe $(b_2, b_1, 1)$ wird $\perp$ zugewiesen (Schritt 2. in DP)
4. $(b_2, b_1, 1)$ zum Zeitpunkt 1 nicht durchführbar

# Stochastische Methoden – GSAT/WalkSAT

*Stochastische Methoden sind unvollständig aber korrekt*

$\longrightarrow$ Restart nach *maxFlips*

$\longrightarrow$ Terminierung nach *maxRestarts*

‘greedy local search” (GSAT)

- Nachbarschaftdefinition (Hamming-Distanz)
- Bewertungsfunktion (# erfüllter Teilklauseln)

“iterative repair” (WALKSAT)

- “Reparieren” einer unerfüllten Klausel

“Random Walk” (WALKSAT)

- drehe zufällige Variable um

# GSAT

**GSAT($\Phi$)**

$\mu \leftarrow$ zufällig
**while** $\mu$ erfüllt $\Phi$ nicht
    **foreach** Variable $V \in \Phi$, $\mu_V \leftarrow \mathit{Flip}(V, \mu)$
    $\mu \leftarrow \mu_P$ mit meisten erfüllten Teilklauseln
**return** $\mu$

- Flip auch wenn Verschlechterung
- keine Erfolgsgarantie, aber in Praxis sehr erfolgreich

# Zusammenfassung

- Planungsproblem der Länge *n* als Axiome
- Expandieren zu Klauselmenge
- Lösen mit *beliebigem* SAT-Solver
- keine Spezialalgorithmen
- zur Zeit schnellster STRIPS-basierter Planer
- und ausdrucksstärker

$$\forall A, B, I.\mathit{frei}(A, I) \equiv \neg\exists B.\mathit{auf}(B, A, I)$$

# Ausblick

- Effizienzsteigerung durch Kodierung
- Repräsentation der Axiome
- geeignetere Frameaxiome
- trotz allem kein voller PDDL Umfang

# Literatur (SAT-Planen)


- Ghallab, M., Nau, D., and Traverso, P. (2004).
  Automated planning : Theory & practice.
- Selman, H. A. K. . B. (1992).
  Planning as satisfiability.
  In *Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI'92)*, pages 359–363.

# Literatur (Algorithmen)


- Cook, S. A. and Mitchell, D. G. (1997).
  Finding hard instances of the satisfiability problem: A survey.
  In Du, Gu, and Pardalos, editors, *Satisfiability Problem: Theory and Applications*, volume 35, pages 1–17. American Mathematical Society.

- Selman, B., Levesque, H. J., and Mitchell, D. (1992).
  A new method for solving hard satisfiability problems.
  In Rosenbloom, P. and Szolovits, P., editors, *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 440–446, Menlo Park, California. AAAI Press.

- Selman, B., Kautz, H. A., and Cohen, B. (1994).
  Noise strategies for improving local search.
  In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94)*, pages 337–343, Seattle.

# Literatur (Kodierung)

Kautz, H. and Selman, B. (1996).
Pushing the envelope: Planning, propositional logic, and stochastic search.

Ernst, M., Millstein, T. D., and Weld, D. S. (1997).
Automatic SAT-compilation of planning problems.
In *IJCAI*, pages 1169–1177.