Satisfiability Planning

Verena Fritzsche
WS 2002/2003

Gliederung

1. Traditioneller Ansatz
2. Neuer Ansatz: Satisfiability Planning
3. Experimentelle Untersuchungen
4. Zusammenfassung
5. Quellen / Literatur

Traditioneller Ansatz

- Planen traditionell als **Deduktion** formalisiert
  → **Plan-Problem beschrieben durch ein Theorem der Form**

  Anfangsbedingungen + Sequenz von Aktionen ⊃ Zielbedingungen

  → **Planen = Finden eines deduktiven Beweises für das Theorem**

- Axiome geben an, dass aus dem Auftreten einer Aktion - bei Erfülltsein ihrer Vorbedingungen - ihre Effekte folgen

  → **z.B.:** $\forall x,y,z,i.\ on\ (x,y,i) \wedge clear\ (x,i) \wedge clear\ (z,i) \wedge move\ (x,y,z,i)$
  $\supset on\ (x,z,i+1) \wedge clear\ (y,i+1)$

- unterschiedliche Repräsentationen für Aktionen und Fluents

Traditioneller Ansatz

- Bsp.:

→ **Formalisierung:**

$$\exists\ x_1, y_1, z_1, x_2, y_2, z_2.$$
$$on\ (A,B,1) \wedge on\ (B,table,1) \wedge clear(A,1)$$
$$\wedge\ move\ (x_1,y_1,z_1,1) \wedge move\ (x_2, y_2, z_2,2) \supset on\ (B,A,3)$$

→ **Plan entspricht dann der Instantiierung der beiden Instanzen des move-Prädikates**

Traditioneller Ansatz

- Verbreiteter Glaube, Planen sein ein von Natur aus systematischer Prozess

  → **systematische Suche im Zustandsraum oder im Raum partieller Pläne**

  → **großer Erfolg des STRIPS-Systems schuf grundlegendes Paradigma für praktisch alle nachfolgende Arbeit im Planen**

Satisfiability Planning

- Neuer Ansatz:

  → **Test auf aussagenlogische Erfüllbarkeit**

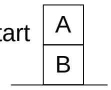
  → **gesamte aufgabenspezifische Information in uniformer Notation repräsentiert (Klauselform)**

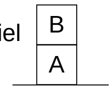
  → **keine Unterscheidung zwischen Operationen und Zuständen (Fluents)**

  Plan-Problem = Menge von Axiomen
  Planen = Finden eines Modells für diese Axiome

  → **zu jedem Modell der Axiome korrespondiert ein Plan**

Satisfiability Planning

- Beispiel von vorhin: Start 

2-Schritt-Plan ? Ziel 

- **Axiomatisierung:**
  - → **Start- und Zielzustand**
    - **on (A,B,1) ∧ on (B,table,1) ∧ clear (A,1) ∧ on (B,A,3)**
  - → **Aktionen**
    - **∀x,y,z,i. on (x,y,i) ∧ clear (x,i) ∧ clear (z,i) ∧ move (x,y,z,i)**
      ⊃ on (x,z,i+1) ∧ clear (y,i+1)
  - → **Frame-Axiome**
    - **∀x,y,u,v,w,i. on (x,y,i) ∧ move (u,v,w,i) ∧ x≠u ⊃ on (x,y,i+1)**
    - **∀x,u,v,w,i. clear (x,i) ∧ move (u,v,w,i) ∧ x≠w ⊃ clear (x,i+1)**

Satisfiability Planning

- **Finden eines Modells für die Axiome:**
  - → **Bsp. 1: on (A,B,1), on (B,table,1), clear (A,1),**
    on (B,A,2), on (B,A,3)
    - → **erfüllt die gegebenen Axiome**
    - → **Aber: Welt ändert sich, obwohl keine bekannte Aktion auftritt**
  - → **Bsp. 2: on (A,B,1), on (B,table,1), clear (A,1),**
    move (B,table,A,1), on (B,A,2), on (B,A,3)
    - → **erfüllt ebenfalls alle Axiome**
    - → **Aber: es wird eine Aktion ausgeführt, deren Vorbedingungen nicht erfüllt sind**

Satisfiability Planning

- Einfache Übernahme der Axiome aus dem deduktiven Ansatz genügt nicht
- unerwünschte Modelle müssen ausgeschlossen werden
- Einführung von zusätzlichen Axiomen:
  - → **Möglichkeit ausschließen, dass Aktionen ausgeführt werden, deren Vorbedingungen falsch sind**

    → eine Aktion impliziert sowohl ihre Vorbedingungen, als auch ihre Effekte

  - → **Bsp.: $\forall x,y,z,i.\ move\ (x,y,z,i) \supset (clear\ (x,i) \wedge clear\ (z,i) \wedge on\ (x,y,i))$**

Satisfiability Planning

→ Zu jedem Zeitpunkt wird nur eine Aktion ausgeführt

→ **Bsp.: $\forall x,x',y,y',z,z',i.\ (x \neq x' \vee y \neq y' \vee z \neq z') \supset \neg\ move\ (x,y,z,i) \vee \neg\ move\ (x',y',z',i)$**

→ Zu jedem Zeitpunkt wird irgendeine Aktion ausgeführt

- **kann auch no-op sein**

→ **Bsp.: $\forall\ i<N.\ \exists x,y,z.\ move\ (x,y,z,i)$**

- Wenn ein Plan-Problem durch Angabe eines kompletten Initial-zustandes spezifiziert wird, so garantieren diese Axiome, dass alle Modelle zu gültigen Plänen korrespondieren

Satisfiability Planning

- Zurück zum Beispiel von vorhin:

→ **einziges Modell:**

on (A,B,1), on (B,table,1), clear(A,1),
move (A,B,table,1), on (A,table,2), clear(B,2),
move (B,table,A,2), on(B,A,3)

Satisfiability Planning

- Vorteil des Ansatzes: sehr ausdrucksstark

→ **beliebige Bedingungen über beliebige (Zwischen-)Zustände und über die Struktur des Planes selbst**

- **z.B. zum Zeitpunkt 5 soll auf Block C oder auf Block D etwas stehen**
  → **¬ clear(C,5) ∨ ¬clear(D,5) zur Problem-Spezifikation hinzufügen**
  → **schwierig im deduktiven Ansatz**

- **z.B. Aktion move (A,B,C,3) soll nicht enthalten sein**
  → **¬ move (A,B,C,3) zur Spezifikation hinzufügen**
  → **ist im deduktiven Ansatz nicht möglich**

Satisfiability Planning

→ Beziehungen zwischen Prädikaten können explizit angegeben werden

- es ist nicht nötig, „einfache“ von „abgeleiteten“ Prädikaten zu unterscheiden
- z.B. behandeln die meisten STRIPS-Operatoren die Prädikate clear und on separat
- im Erfüllbarkeitsansatz kann einfach zugewiesen werden:
  clear (x,i) $\equiv \neg\exists$y. on (y,x,i)

Satisfiability Planning

- Zusammenfassung:
  - Plan-Problem = Menge von Axiomen
  - Planen = Finden eines Modells für diese Axiome
  - keine Unterscheidung zwischen Axiomen für Operationen und Axiomen für Zustände
  - Domain-Axiome im allg. stärker als im deduktiven Ansatz, da ungewollte Modelle ausgeschlossen werden müssen:

    → 1. Eine Aktion impliziert sowohl ihre Vorbedingungen, als auch ihre Effekte
    → 2. Zu jedem Zeitpunkt wird genau eine Aktion ausgeführt
    → 3. Anfangszustand ist vollst~~ändig spez~~ifiziert

  - jedes Modell der Axiome korrespondiert zu einem gültigen Plan

Experimentelle Untersuchungen

- Vergleich von verschiedenen Problem - Encodings

  → **1. Graphplan-basierte Darstellung**
  → **2. Lineare Darstellung**
  → **3. Zustandsbasierte Darstellung**

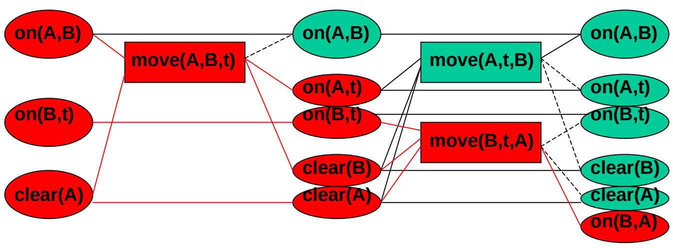
- Systematische vs. Stochastische Suche

- Betrachtung von Instanzen aus der rocket- und der logistics-Domäne, sowie einiger großer Blockwelt-Probleme

Experimentelle Untersuchungen
Problem - Encodings

Graphplan-basierte Darstellung

- wird gewonnen aus dem Plangraphen



Experimentelle Untersuchungen
Problem - Encodings

Graphplan-basierte Darstellung (2)

- ~~Plangraph kann in KNF konvertiert~~ werden:
- Startzustand = gültige Fluents in Schicht 1
  Zielzustand = gültige Fluents in höchster Schicht

  → **on (A,B,1) ∧ on (B,table,1) ∧ clear (A,1) ∧ on (B,A,3)**
- Durchlaufen des Plangraphen von Zielschicht rückwärts

  → **jede Proposition in Level i impliziert die Disjunktion aller Aktionen des Levels i-1, die sie als add-Effekt haben**
  z. B. on (A,B,3) ⊃ (no-op (on (A,B,2)) ∨ move (A,table,B,2))
  → **Aktionen implizieren ihre Vorbedingungen**
  z. B. move (B,table,A,2) ⊃ (clear (B,2) ∧ clear (A,2) ∧ on (B,table,2))
  → **widersprüchliche Aktionen schließen sich wechselseitig aus**
  z. B. ¬ move (A,table,B) ∨ ¬ move (B,table,A)
- liefert Klausel-Form des Erfüllbarkeitsansatzes

Experimentelle Untersuchungen
Problem - Encodings

Lineare Darstellung

- **Bedingungen, um sicherzustellen, dass alle Modelle für eine Menge von Axiomen zu gültigen Plänen korrespondieren:**

  → **eine Aktion impliziert sowohl ihre Vorbedingung als auch ihre Effekte**
  → **zu jedem Zeitpunkt tritt genau eine Aktion auf**
- **Modelle korrespondieren unter diesen Bedingungen zu linearen Plänen**
- **Problem: viele Aktionen → Darstellung sehr lang**

Experimentelle Untersuchungen
Problem - Encodings

Lineare Darstellung (2)

- **größte Reduktion der Problem-Größe durch Reduzieren der Quantifier-Schachtelungstiefe erreichbar**

Bsp.: $\forall$ x,x‘,y,y‘,z,z‘,i. (x≠x‘ $\vee$ y≠y‘ $\vee$ z≠z‘)
$\supset \neg$ move (x,y,z,i) $\vee$ $\neg$ move (x‘,y‘,z‘,i)

move (x,y,z,i) ersetzen durch {object (x,i), source (y,i), dest (z,i)}

$\forall i, x_1,x_2.\ x_1 \neq x_2 \supset \neg\, \text{object}\ (x_1,i) \vee \neg\, \text{object}\ (x_2,i)$
$\forall i, y_1,y_2.\ y_1 \neq y_2 \supset \neg\, \text{source}\ (y_1,i) \vee \neg\, \text{source}\ (y_2,i)$
$\forall i, z_1,z_2.\ z_1 \neq z_2 \supset \neg\, \text{dest}\ (z_1,i) \vee \neg\, \text{dest}\ (z_2,i)$

Experimentelle Untersuchungen
Problem - Encodings

Zustandsbasierte Darstellung

- Zustandsaxiome geben an, was es für jeden individuellen Zustand heißt, gültig zu sein
  - → **z. B. in der blocks world sichern Zustandsaxiome, dass auf einem Block nur ein anderer sein kann; dass ein Block nicht gleichzeitig frei sein und etwas auf ihm stehen kann, usw. ...**
- Zustandsaxiome sichern, dass jeder Zustand intern konsistent ist
- zur Beschreibung von Zustandsübergängen relativ wenige Axiome nötig
  - → **Axiome über mögliche Aktionen, die für eine Änderung verantwortlich sein können**
    - z. B. ($\neg$ on (A,B,i) $\wedge$ on (A,B,i+1)) $\supset \exists$ z . move (A,z,B,i)
  - → **Axiome über den wechselseitigen Ausschluss von widersprüchlichen Aktionen**
  - → **Axiome, die zusichern, dass eine Aktion ihre Vorbedingungen und ihre Effekte impliziert**

Experimentelle Untersuchungen
Problem - Encodings

- **Kann noch kompakter gemacht werden:**

  → **1. Trick mit der Argumentreduktion**
  → **2. Eliminierung von Aktions-Axiomen, da ihre Wirkung mittels Zustandsaxiomen ausgedrückt werden kann**

  → **Bsp.: Logistik-Domäne**

  Ersatz expliziter load-truck-, load-airplane-Axiome durch Zustandsaxiom:
  at (obj,loc,i) $\supset$ at (obj,loc,i+1) $\vee$ $\exists$x $\in$ truck $\cup$ airplane.
  in (obj,x,i+1) $\wedge$ at (x,loc,i) $\wedge$ at (x,loc,i+1)

  - **sogar soweit fortführbar, dass nur Axiome über Fluents übrigbleiben**

Experimentelle Untersuchungen
Verwendete Algorithmen

Tableau

- **systematischer Erfüllbarkeitstester**
- **baut inkrementell Wahrheitswert-Zuweisungen auf**
- **geht zurück, sobald er feststellt, dass die Zuweisung die Formel nicht erfüllt**
- **Vereinfachung der Formel während des Durchlaufens**
- **einer der momentan schnellsten vollständigen Erfüllbarkeitstester**

Experimentelle Untersuchungen
Verwendete Algorithmen

Walksat

- basiert auf zufallsinitialisierter, lokaler Greedy-Suche (Stochastische Methode)
- Wahl einer zufälligen Wahrheitswertebelegung für die Formel
- Zufällige Wahl einer Klausel, die von der Belegung nicht erfüllt wird
- Wahrheitswert einer Variablen dieser Klausel wird geändert („flip“) und damit die Klausel erfüllt
- mehrere andere Klauseln können durch den „flip“ unerfüllt werden

Experimentelle Untersuchungen
Verwendete Algorithmen

Walksat (2)

- greedy-Strategie: Auswahl der Variablen, bei deren Änderung die sich ergebende Anzahl erfüllter Klauseln maximal ist
- Strategie zufällig mit einer gewissen Häufigkeit angewendet, um lokale Extrema zu umgehen
- Algorithmus „flipt“ solange, bis erfüllende Belegung gefunden oder vordefiniertes Maximum an „flips“ erreicht
- wenn keine erfüllende Belegung gefunden, Neustart mit anderer Zufallsinitialisierung
- von Natur aus unvollständig, d. h., kann nicht beweisen, dass eine Formel unerfüllbar ist

Experimentelle Untersuchungen
Ergebnisse

Problem	Zeit / Aktionen	Graphplan		SAT-Darstellung					
				Graphplan-basiert			Direkt		
		Knoten	Zeit	Variablen	syst.	stoch.	Variablen	syst.	stoch
rocket_ext.a	7/34	1.625	520	1.103	4,4	4,7	331	0,8	0,1
rocket_ext.b	7/30	1.701	2.337	1.179	2,8	21	351	2,5	0,2
logistics.a	11/54	2.891	6.743	1.782	6,9	29	828	--	2,7
logistics.b	13/47	3.382	2.893	2.069	6,4	47	843	--	1,6
logistics.c	13/65	4.326	--	2.809	23.061	262	1.141	--	1,9
bw_large.a	6/12	5.779	11,5	5.772	--	--	459	0,5	0,3
bw_large.b	9/18	18.069	27.115		--	--	1.087	1,5	22
bw_large.c	14/28		--		--	--	3.016	564	670
bw_large.d	18/36		--		--	--	6.764	--	937

Experimentelle Untersuchungen
Ergebnisse

- Direkte (lineare / zustandsbasierte) Darstellung am kompaktesten
  → am schnellsten gelöst
- Graphplan-basierte Darstellung + Erfüllbarkeitsalgorithmus leistungsfähiger als Graphplan-System selbst
- stochastische Suche übertrifft systematische in den meisten Fällen deutlich
- besonders gute Performance bei Walksat + zust.basierter Darstellg.
  → Stochastische Methoden + effiziente Darstellungstechniken vielversprechender Ansatz

Experimentelle Untersuchungen
Ergebnisse

- Systematische und stochastische Methoden ergänzen sich:

  → z.B. Instanz bw_large.d:

  - Walksat + direkte Darstellung findet den optimalen Plan
  - Tableau findet keine Lösung, kann aber beweisen, dass es keinen kürzeren Plan gibt

  → systematsche und stochastische Methoden kombinierbar:

  - stochastische Methoden zur Plan-Synthese
  - systematische Methoden zum Finden unterer Schranken für die Plan-Länge

Zusammenfassung

- Modell für das Planen basierend auf Erfüllbarkeit anstatt auf Deduktion
- Verstärkung der deduktiven Axiome nötig, um ungewollte Modelle auszuschließen
- sehr ausdrucksstark
- verschiedene effiziente Problem-Encodings

  → **zustandsbasierte Darstellung am kompaktesten**

- Einsatz von stochastischen Methoden

  → **sind den systematischen oft stark überlegen**

- Kombination von stochastischen und systematischen Methoden

Quellen / Literatur

- Kautz, H. & Selman, B. - Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search
- Kautz, H. & Selman, B. - Planning as Satisfiability
- Prof. G. Brewka - Materialien zur Vorlesung WBS - WS 01/02
- Cohen, P. & Feigenbaum, E. - The Handbook of Artificial Intelligence