Grundlagen der Verarbeitung von Wissen über Raum, Zeit und Ereignisse

Carola Eschenbach, Christopher Habel
Universität Hamburg, FB Informatik
AB Wissens- und Sprachverarbeitung (WSV)

Grundlagen der Verarbeitung von Wissen über Raum, Zeit und Ereignisse

Wissen über Aktionen

- Situationskalkül
  - Spezifikationen von Aktionen
  - Frameproblem
  - Planung

Hintergrundwissen über Handlungen

für Aktionsauswahl, Planung

Erforderlich

- Voraussetzungen der Handlung
- Effekte der Handlung

Nützlich

- Kosten (Zeit, ...)
- Verlässlichkeit

Wissen über Aktionen, Handlungen

Legalitätsfeststellung

- Ist eine Aktion in einer gegebenen Situation ausführbar ?
- Ist eine Aktionsfolge ausführbar ? (Wenn ja: In welchen Situationen ?)

Vorhersage

- In welchen Weltzustand führt die Aktion / Aktionsfolge von einer gegebenen Situation?

Aktionswahl

- Welche Aktion bringt mich meinem Ziel näher?

Planung

- Welche Aktionsfolge führt in einen Weltzustand, der meiner Zielbeschreibung entspricht ?

Zeitabhängigkeit

Repräsentationen von Wissen enthalten

- zeitunabhängige und
- zeitabhängige Anteile

Unterscheidungsmöglichkeiten

- Trennung in der ‚Architektur' (STRIPS)
- Zeitbezug in den Formeln explizit machen

Aktionen / Handlungen

- ‚treiben die Zeit voran'
- ‚bewirken, welcher Zustand einem vorherigen folgt'

Handlungsbeschreibung in STRIPS (1971)

Aktionen als Produktionen

- Name: pickup(x)
- LHS: **preconditions**: Konjunktion von atomaren Formeln
  - ONTABLE(x) ∧ HANDEMPTY ∧ CLEAR(x)
- RHS: **Delete** list: atomare Formeln:
  - ONTABLE(x), HANDEMPTY, CLEAR(x)
- RHS: **Add** formula: atomare Formel: HOLDING(x)

Beispiel: Produktionsausführung in STRIPS

Faktenbasis (Zustand 1)
ONTABLE(A)
ONTABLE(B)
HANDEMPTY
CLEAR(A)
ON(C, B)

pickup(A)

Faktenbasis (Zustand 2)
ONTABLE(B)
ON(C, B)
HOLDING(A)

- Die Zustände 1 und 2 der Faktenbasis repräsentieren verschiedene Weltzustände
- Es besteht keine (einfache) logische Folgerungsbeziehung.

Beispiel: Situationskalkül

Faktenbasis
ONTABLE(A, s_0)
HANDEMPTY(s_0)
CLEAR(A, s_0)

ONTABLE(B, s_0)
ON(C, B, s_0)

s_1 = do(pickup(A), s_0)

Ergänzung der Faktenbasis
HOLDING(A, s_1)
¬ONTABLE(A, s_1)
¬HANDEMPTY(s_1)
¬CLEAR(A, s_1)

- s_0 und s_1 repräsentieren verschiedene Weltzustände
- Die Handlungsbeschreibungen können nicht alles explizit machen, was die Handlung unverändert lässt

Handlungsbeschreibung im Situationskalkül

Spezifikationen von Aktionen

Name: pickup(x)

preconditions

Poss(a, s): a kann in s ausgeführt werden

$\forall x, s\ [Poss(pickup(x), s) \Leftrightarrow ONTABLE(x, s) \wedge$
$HANDEMPTY(s) \wedge CLEAR(x, s)]$

effects

do(a, s): Bezeichnung der Situation, die das Ausführen von a in s herbeiführt

$\forall x, s\ [\neg ONTABLE(x, do(pickup(x), s)) \wedge$
$\neg HANDEMPTY(do(pickup(x), s)) \wedge$
$\neg CLEAR(x, do(pickup(x), s)) \wedge$
$HOLDING(x, do(pickup(x), s))]$

Situationskalkül

Basisannahmen

- Nur der Agent verändert durch seine Aktionen die Welt
  - keine Berücksichtigung der Wahrnehmung
- Zeit: nur implizit über Aktionsfolgen
  - keine Dauer
  - keine kontinuierlichen Prozesse
  - keine Parallelität

Ziel

- Aktionsauswahl
- Planung
- Vorhersage

Aufhebung der Annahmen

- durch übergeordnete Mechanismen: z.B. Golog

Situationskalkül: Prädikatenlogik mit

neuen Sorten von Objekten / Terme für

- (primitive) **Aktionen**: Anwendungsabhängige **Typen**
- **Situationen**: mögliche Historien (Initialzustand der Welt und Sequenzen von Aktionen davon ausgehend)

Prädikaten und Relationssymbolen

- ohne Situationsargument
  - unveränderliche Eigenschaften, Beziehungen heavy(x)
- mit Situationsargument (Konvention: an letzter Position)
  - veränderliche Eigenschaften Beziehungen ontable(x, s)
  - **Fluents**
- Ausgezeichneter Fluent
  - Poss(a, s): In Situation s kann a ausgeführt werden
  - zur Spezifikation von Vorbedingungen von Aktionen

Situationskalkül: (syntaktische) Varianten

Expliziter oder impliziter Agent

- Aktionsterme können den Agenten explizit benennen
  Poss(pickup(r, x), s) ⇔ ¬heavy(x) ∧ nextTo(r, x, s)
- ➔ Aktionsterme können den Agenten implizit lassen (wenn es nur einen gibt)
  Poss(pickup(x), s) ⇔ ¬heavy(x) ∧ nextTo(x, s)

Fluents als Terme

- werden mit ausgezeichneter Relation Holds zu Situationen in Beziehung gesetzt
  Poss(pickup(x), s) ⇔ ¬heavy(x) ∧ Holds(nextTo(x), s)
- erfordern die Duplizierung logischer Symbole für die Bildung komplexer Fluents
  Holds(Φ & Ψ, s) ⇔ Holds(Φ, s) ∧ Holds(Ψ, s)

Wissen über Aktionen im Situationskalkül

Legalitätsfeststellung

- Ist Aktion a in Situation s ausführbar ?
  $\mathcal{KB} \models$ Poss(a, s)

Vorhersage

- Gilt Fluent Φ nach Ausführung von Aktion a in Situation s?
  $\mathcal{KB} \models \Phi$(do(a, s))
  bzw.
  $\mathcal{KB} \models$ Holds(Φ, do(a, s))

Verallgemeinerbar

- Für Aktionssequenzen

Wissen über Aktionssequenzen im Situationskalkül

Es sei

$\langle a_1, \ldots, a_n \rangle$ eine Sequenz von primitiven Aktionen

Kurzschreibweisen

$do(\langle\rangle, s) = s$

$do(\langle a_1, \ldots, a_n \rangle, s) = do(a_n, do(\langle a_1, \ldots, a_{n-1} \rangle, s))$

$Legal(\langle\rangle, s) = \top$

$Legal(\langle a_1, \ldots, a_n \rangle, s) = Legal(\langle a_1, \ldots, a_{n-1} \rangle, s) \wedge Poss(a_n, do(\langle a_1, \ldots, a_{n-1} \rangle, s))$

Legalitätsfeststellung

Ist $\langle a_1, \ldots, a_n \rangle$ in Situation s ausführbar ?

$\mathcal{KB} \models Legal(\langle a_1, \ldots, a_n \rangle, s)$

Vorhersage

Gilt Fluent Φ nach Ausführung von $\langle a_1, \ldots, a_n \rangle$ in s?

$\mathcal{KB} \models \Phi(do(\langle a_1, \ldots, a_n \rangle, s))$

Aufzug

(Levesque et al. 1997, mit Ergänzungen)

Primitive actions

turnoff(n)	% Turn off call button n.
open	% Open elevator door.
close	% Close elevator door.
up(n)	% Move elevator up to floor n.
down(n)	% Move elevator down to floor n.

Fluents

on(n, s)	% unanswered call on floor n
currentFloor(n, s)	% current floor has number n
+d_closed(s)	% the elevator door is closed

Aufzug: Vorbedingungen

Preconditions for Primitive Actions

$\mathcal{Pre}_{EL}$

∀s, n [poss(turnoff(n),s) ⇔ on(n, s) ∧ currentFloor(n, s) ∧ ¬d_closed(s)]

% Anforderung befriedigt wenn mit offener Tür auf Etage

∀s [poss(open,s) ⇔ d_closed(s)] % nur geschl. Tür öffnen

∀s [poss(close,s) ⇔ ¬d_closed(s)] % nur offene Tür schließen

∀s, n, m [poss(up(n), s) ⇔ currentFloor(m, s) ∧ m < n ∧ d_closed(s)] %Fahrt nur mit geschlossener Tür

∀s, n, m [poss(down(n), s) ⇔ currentFloor(m, s) ∧ m > n ∧ d_closed(s)] %Fahrt nur mit geschlossener Tür

Aufzug: Effekte

Effect Axioms for Primitive Fluents

$\mathcal{Eff}_{EL}$

Positive Effekte

$\forall$s [d_closed(do(close, s))] % Tür zu

$\forall$s, m [currentFloor(m, do(up(m), s))] % Position

$\forall$s, m [currentFloor(m, do(down(m), s))] % Position

Negative Effekte

$\forall$s, m [¬on(m, do(turnoff(m), s))] % Anforderung aus

$\forall$s [¬d_closed(do(open, s))] % Tür auf

$\forall$s,n,m [n≠m ⇒ ¬currentFloor(m,do(up(n),s))] % Position

$\forall$s,n,m [n≠m ⇒ ¬currentFloor(m,do(down(n),s))] % Position

Mögliche Aktionen

Initiale Situation s_0

Call buttons: 3 and 5. Elevator is at floor 4. Door is open.

Init_{EL}: on(3, s_0), on(5, s_0), currentFloor(4, s_0)

Mögliche Aktionen in

s_0	do(close, s_0)	do(up(5), do(close, s_0))	do(open, do(up(5), do(close, s_0)))
close	open up(5) down(3) down(2) down(1)	open down(4) down(3) down(2) down(1)	close turnoff(5)

?

$\mathcal{Pre}_{EL} \cup \mathcal{Eff}_{EL} \cup \mathcal{Init}_{EL} \vDash$ **poss(close, s_0) ?**

$\forall s$ [poss(close, s) $\Leftrightarrow$ $\neg$d_closed(s)] $\in \mathcal{Pre}_{EL}$

Initiale Situation

- Call buttons: 3 and 5. Elevator is at floor 4. Door is open.

on(3, s_0), on(5, s_0), currentFloor(4, s_0) $\in \mathcal{Init}_{EL}$

- Closed World Assumption (?)

Vollständige Spezifikation der initialen Situation

$\mathcal{Init}_{EL}$

$\forall$ m [on(m, s_0) $\Leftrightarrow$ m = 3 $\vee$ m = 5]

$\forall$ m [currentFloor(m, s_0) $\Leftrightarrow$ m = 4]

$\neg$d_closed(s_0)

$\mathcal{Pre}_{EL} \cup \mathcal{Eff}_{EL} \cup \mathcal{Init}_{EL} \vDash$ **poss(open, do(close, s_0)) ?**

$\forall s$ [poss(open, s) $\Leftrightarrow$ d_closed(s)] $\in \mathcal{Pre}_{EL}$

Positives Effekt-Axiom

$\forall s$ [d_closed(do(close, s))] $\in \mathcal{Eff}_{EL}$

Also gilt sogar

- $\mathcal{Pre}_{EL} \cup \mathcal{Eff}_{EL} \vDash$ $\forall s$ [poss(open, do(close, s))]
- (Was auch immer ist, wenn man gerade die Tür geschlossen hat, dann kann man sie auch wieder öffnen)

$\mathcal{Pre}_{EL} \cup \mathcal{Eff}_{EL} \cup \mathcal{Init}_{EL} \models$ poss(up(5), do(close, s_0)) ?

$\forall$s, n, m [poss(up(n), s) $\Leftrightarrow$ currentFloor(m, s) $\wedge$
m < n $\wedge$ d_closed(s)] $\in \mathcal{Pre}_{EL}$

Gilt $\mathcal{Pre}_{EL} \cup \mathcal{Eff}_{EL} \cup \mathcal{Init}_{EL} \models$ currentFloor(m, do(close, s_0)) für irgendein m?

Initiale Situation

$\forall$ m [currentFloor(m, s_0) $\Leftrightarrow$ m = 4] $\in \mathcal{Init}_{EL}$

Effekt-Axiome zu currentFloor: $\mathcal{Eff}_{EL}$

$\forall$s, m [currentFloor(m, do(up(m), s))]

$\forall$s, n, m [n $\neq$ m $\Rightarrow$ $\neg$currentFloor(m, do(up(n), s))]

$\forall$s, m [currentFloor(m, do(down(m), s))]

$\forall$s, n, m [n $\neq$ m $\Rightarrow$ $\neg$currentFloor(m, do(down(n), s))]

Das Schließen der Tür hat keinen Effekt auf die Position des Aufzugs, aber wo steht das?

Frame-Problem

McCarthy & Hayes (1969)

Effekte von Handlungen

- sind (in der Regel) beschränkt
  - Das Fallenlassen des Stiftes verändert nicht seine Farbe.
  - Das Schließen der Tür verändert nicht die Position des Aufzugs.
- und die Beschränkungen sind uns (weitgehend) bekannt.

Frame-Axiome

- machen Aussagen über Invarianz von Fluents
- (Kein-)Effekt-Axiome
- Sind im Normalfall nicht aus Effekt-Axiomen folgerbar.

Im Aufzug-Beispiel

- Das Öffnen und Schließen der Tür und das Ausschalten der Anforderung verändert die Position des Fahrstuhls nicht.

$\mathcal{Fr}_{EL}$

$\forall$s,m,n [currentFloor(m,do(turnoff(n),s)) $\Leftrightarrow$ currentFloor(m,s)]

$\forall$s,m [currentFloor(m,do(open,s)) $\Leftrightarrow$ currentFloor(m,s)]

$\forall$s,m [currentFloor(m,do(close,s)) $\Leftrightarrow$ currentFloor(m,s)]

$\mathcal{Pre}_{EL} \cup \mathcal{Eff}_{EL} \cup \mathcal{Fr}_{EL} \cup \mathcal{Init}_{EL} \vDash$ poss(up(5), do(close, s_0))

$\forall$s, n, m [poss(up(n), s) $\Leftrightarrow$ currentFloor(m, s) $\wedge$
m < n $\wedge$ d_closed(s)] $\in \mathcal{Pre}_{EL}$

Gilt $\mathcal{Pre}_{EL} \cup \mathcal{Eff}_{EL} \cup \mathcal{Fr}_{EL} \cup \mathcal{Init}_{EL} \vDash$
currentFloor(m, do(close, s_0)) für irgendein m?

Initiale Situation

$\forall$m [currentFloor(m, s_0) $\Leftrightarrow$ m = 4] $\in \mathcal{Init}_{EL}$

Frame Axiom

$\forall$s,m[currentFloor(m,do(close,s))$\Leftrightarrow$currentFloor(m,s)] $\in \mathcal{Fr}_{EL}$

Also gilt

$\mathcal{Fr}_{EL} \cup \mathcal{Init}_{EL} \vDash$ currentFloor(4, do(close, s_0))

$\mathcal{Pre}_{EL} \cup \mathcal{Eff}_{EL} \cup \mathcal{Fr}_{EL} \cup \mathcal{Init}_{EL} \vDash$ poss(up(5), do(close, s_0))

Weitere Frame-Axiome

$\mathcal{Eff}_{EL}$

$\forall s, m\ [\neg on(m, do(turnoff(m), s))]$

$\forall s\ [d\_closed(do(close, s))]$

$\forall s\ [\neg d\_closed(do(open, s))]$

$\mathcal{Fr}_{EL}$

$\forall s, m\ [on(m,do(open,s)) \Leftrightarrow on(m,s)]$

$\forall s, m\ [on(m,do(close,s)) \Leftrightarrow on(m,s)]$

$\forall s, m, n\ [on(m,do(up(n),s)) \Leftrightarrow on(m,s)]$

$\forall s, m, n\ [on(m,do(down(n),s)) \Leftrightarrow on(m,s)]$

$\forall s, m\ [d\_closed(do(turnoff(m),s)) \Leftrightarrow d\_closed(s)]$

$\forall s, m\ [d\_closed(do(up(m),s)) \Leftrightarrow d\_closed(s)]$

$\forall s, m\ [d\_closed(do(down(m),s)) \Leftrightarrow d\_closed(s)]$

Frame-Axiome

Mögliches Problem

- Kein-Effekt ist der Normalfall.
- Bei n primitiven Aktionen und m primitiven Fluents ergeben sich O(m*n) Frame-Axiome
- (Wie) lassen sich die Frame-Axiome automatisch bestimmen?

Closed World Assumption für Fluents und Aktionen

- Wenn ein Fluent durch eine Aktion nicht verändert wird, dann hat er nach der Durchführung denselben Wert wie vorher.
- Bei vollständigem Wissen über die Effekte aller primitiven Aktionen ist die Bestimmung der Frame-Axiome möglich.

Bestimmung der Frame-Axiome

Gegeben

- ein Fluent Φ
- Vollständige Spezifikation der positiven und negativen Effekte aller Handlungen bzgl. Φ

Ziel

- Generierung der Frame-Axiome für Φ
- Darstellung in kompakter Form

Lösung

- Zusammenfassung der Effekt-Axiome
- Generierung der Erklärungsabschlussaxiomen (explanation closure)
- Bestimmung von Nachfolgezustandsaxiomen (successor state axioms)

Bestimmung der Frame-Axiome: Vorbereitung

- bringe die positiven und negativen Effekt-Axiome zu Fluent Φ in folgende Normalform,
  $\forall x, s\ [\Psi_i(x, s) \Rightarrow \Phi(x, do(a_i, s))]$
  $\forall x, s\ [\Theta_j(x, s) \Rightarrow \neg\Phi(x, do(b_j, s))]$
  s eine Situationsvariable, x ein Variablen-Vektor, a_i, b_j Namen von Aktionen, $\Psi_i(x, s)$, $\Theta_j(x, s)$ Formeln mit x und s als freie Variablen
- Schreibe die Normalformen um (mit a, b neue Variablen)
  $\forall x, s, a\ [\Psi_i(x, s) \wedge a = a_i \Rightarrow \Phi(x, do(a, s))]$
  $\forall x, s, b\ [\Theta_j(x, s) \wedge b = b_j \Rightarrow \neg\Phi(x, do(b, s))]$
- Fasse positive bzw. negative Effekte zusammen zu
  $\forall x, s, a\ [\Pi_\Phi(x, a, s) \Rightarrow \Phi(x, do(a, s))]$
  $\forall x, s, b\ [N_\Phi(x, b, s) \Rightarrow \neg\Phi(x, do(b, s))]$

Beispiel: currentFloor(x, s)

- Effekt-Axiome: $\forall$s, x [currentFloor(x, do(up(x), s))]
  $\forall$s, x [currentFloor(x, do(down(x), s))]
  $\forall$s, n, x [n $\neq$ x $\Rightarrow$ $\neg$currentFloor(x, do(up(n), s))]
  $\forall$s, n, x [n $\neq$ x $\Rightarrow$ $\neg$currentFloor(x, do(down(n), s))]
- Normalform
  $\forall$x, s [$\top$ $\Rightarrow$ currentFloor(x, do(up(x), s))]
  $\forall$x, s [$\top$ $\Rightarrow$ currentFloor(x, do(down(x), s))]
  $\forall$x,n, s [n $\neq$ x $\Rightarrow$ $\neg$currentFloor(x, do(up(n), s))]
  $\forall$x,n, s [n $\neq$ x $\Rightarrow$ $\neg$currentFloor(x, do(down(n), s))]
- Schreibe die Normalformen um
  $\forall$x, s, a [a = up(x) $\Rightarrow$ currentFloor(x, do(a, s))]
  $\forall$x, s, a [a = down(x) $\Rightarrow$ currentFloor(x, do(a, s))]
  $\forall$x, s, b [$\exists$n [n $\neq$ x $\wedge$ b = up(n)] $\Rightarrow$ $\neg$currentFloor(x,do(b,s))]
  $\forall$x, s, b [$\exists$n [n$\neq$x $\wedge$ b=down(n)] $\Rightarrow$ $\neg$currentFloor(x,do(b,s))]

Beispiel: currentFloor(x, s) (Forts.)

- Fasse positive bzw. negative Effekte zusammen zu
  $\forall$x, s, a [Π_Φ(x, a, s) $\Rightarrow$ Φ(x, do(a, s))]
  $\forall$x, s, a [(a=up(x) $\vee$ a=down(x)) $\Rightarrow$ currentFloor(x,do(a,s))]
  $\forall$x, s, b [N_Φ(x, b, s) $\Rightarrow$ $\neg\Phi$(x, do(b, s))]
  $\forall$x, s, b [$\exists$n [(n $\neq$ x $\wedge$ b = up(n)) $\vee$ (n $\neq$ x $\wedge$ b = down(n))]
  $\Rightarrow$ $\neg$currentFloor(x, do(b, s))]

Bestimmung der Frame-Axiome: Erklärungsabschluss

zusammengefasste Effekt-Axiome

$\forall x, s, a\, [\Pi_\Phi(x, a, s) \Rightarrow \Phi(x, do(a, s))]$
$\forall x, s, b\, [N_\Phi(x, b, s) \Rightarrow \neg\Phi(x, do(b, s))]$

Annahme

- Alle Effekte sind explizit kodiert.

Erklärungsabschlussaxiome (explanation closure) $\mathcal{Fr}$

- Erkläre jeden Wechsel durch bekannte Effekte
  $\forall x, s, a\, [\neg\Phi(x, s) \wedge \Phi(x, do(a, s)) \Rightarrow \Pi_\Phi(x, a, s)]$
  $\forall x, s, b\, [\Phi(x, s) \wedge \neg\Phi(x, do(b, s)) \Rightarrow N_\Phi(x, b, s)]$
- in Frame-Axiom-Form: Fehlen Voraussetzungen für Effekte, dann bleibt alles wie gehabt
  $\forall x, s, a\, [\neg\Phi(x, s) \wedge \neg\Pi_\Phi(x, a, s) \Rightarrow \neg\Phi(x, do(a, s))]$
  $\forall x, s, b\, [\Phi(x, s) \wedge \neg N_\Phi(x, b, s) \Rightarrow \Phi(x, do(b, s))]$

Beispiel: currentFloor(x, s)

Bestimmung der Frame-Axiome: Erklärungsabschluss

- zusammengefasste Effekt-Axiome
  $\forall x, s, a\, [(a=up(x) \vee a=down(x)) \Rightarrow currentFloor(x,do(a,s))]$
  $\forall x,s,b[\exists n\, [n\neq x \wedge (b=up(n) \vee b=down(n))] \Rightarrow \neg cFl(x,do(b,s))]$

Erklärungsabschlussaxiome $\mathcal{Fr}_{EL}$

$\forall x, s, a\, [\neg currentFloor(x, s) \wedge currentFloor(x, do(a, s)) \Rightarrow$
$(a = up(x) \vee a = down(x))]$
$\forall x, s, b\, [currentFloor(x, s) \wedge \neg currentFloor(x, do(b, s)) \Rightarrow$
$\exists n\, [n \neq x \wedge (b = up(n) \vee b = down(n))]]$

- in Frame-Axiom-Form
  $\forall x,s,a\, [\neg cFl(x,s) \wedge a\neq up(x) \wedge a\neq down(x) \Rightarrow \neg cFl(x,do(a,s))]$
  $\forall x,s,b\, [cFl(x,s) \wedge \forall n\, [n=x \vee (b\neq up(n) \wedge b\neq down(n))] \Rightarrow$
  $cFl(x,do(b,s))]$

Axiome des Nachfolgezustands

zusammengefasste Effekt-Axiome

$\forall$s, x, a $[\Pi_\Phi$(x, a, s) $\Rightarrow \Phi$(x, do(a, s))]
$\forall$s, x, b $[N_\Phi$(x, b, s) $\Rightarrow \neg\Phi$(x, do(b, s))]

Annahme

- Alle Effekte sind explizit kodiert.
- Integrität: $\mathcal{KB} \models \forall$s, x, a $[\neg(\Pi_\Phi(x, a, s) \wedge N_\Phi(x, a, s))]$

Erklärungsabschlussaxiome (explanation closure)

$\forall$s, x, a $[\neg\Phi(x, s) \wedge \Phi(x, do(a, s)) \Rightarrow \Pi_\Phi(x, a, s)]$
$\forall$s, x, b $[\Phi(x, s) \wedge \neg\Phi(x, do(b, s)) \Rightarrow N_\Phi(x, b, s)]$

Axiom des Nachfolgezustands (successor state axiom)

- Fasst Effekt-Axiome und Erklärungsabschluss zusammen
  $\forall$s,x,a $[\Phi(x,do(a,s)) \Leftrightarrow \Pi_\Phi(x,a,s) \vee (\Phi(x,s) \wedge \neg N_\Phi(x,a,s))]$

Axiome des Nachfolgezustands

Axiom des Nachfolgezustands (successor state axiom)

$\forall$s,x,a $[\Phi(x,do(a,s)) \Leftrightarrow (\Pi_\Phi(x,a,s) \vee (\Phi(x,s) \wedge \neg N_\Phi(x,a,s)))]$

Beispiel: on(x, s)

$\forall$s,x,a [on(x,do(a,s)) $\Leftrightarrow$ (on(x,s) $\wedge$ a $\neq$ turnoff(x))]

Beispiel: d_closed(s)

$\forall$s,a [d_closed(do(a,s))$\Leftrightarrow$(a=close$\vee$(d_closed(x,s)$\wedge$a$\neq$open))]

Beispiel: currentFloor(x, s)

$\forall$s,x,a [currentFloor(x,do(a,s)) $\Leftrightarrow$ (a = up(x) $\vee$ a = down(x) $\vee$
(currentFloor(x,s) $\wedge$ $\neg\exists$n [n$\neq$x $\wedge$ (a=up(n) $\vee$ a=down(n))]))]

Axiome im Situationskalkül

Vorbedingungsaxiome *Pre*

- Für jede Aktion genau eine Äquivalenz
  $\forall s,a\ [Poss(a, s) \Leftrightarrow \Psi(s)]$

Axiome des Nachfolgezustands *SSt*

- Zusammenfassung von
  Effekt-Axiomen *Eff*
  Frame-Axiomen *Fr*
- Für jeden primitiven Fluent genau eine Äquivalenz
  $\forall s,x,a\ [\Phi(x,do(a,s)) \Leftrightarrow \Pi_{\Phi}(x,a,s) \vee (\Phi(x,s) \wedge \neg N_{\Phi}(x,a,s))]$

Spezifikation des Initialzustands *Init*

- s_0 als einziger Situationsausdruck

Lösung des Frame-Problems im Situationskalkül

Vollständige Spezifikation von (primitiven) Aktionen

- Vorbedingungen
- Effekte
  - Integrität (keine inkonsistenten Effekte)
  - Determiniertheit (keine disjunktiven Effekte)

Spezifikation der (primitiven) Fluents

- Nachfolgezustandsaxiome generiert aus Effektspezifikationen der Aktionen (und Vollständigkeitsannahme)
- Keine Abhängigkeiten

(Annahme der eindeutigen Benennung von Aktionen)

- Brachman und Levesque nennen sie als wesentlich, ohne den Grund anzugeben.

Planung im Situationskalkül

Gegeben

- Zielspezifikation: Eine Formel $G(s)$ mit s als einziger freien Variable
- Startsituation: eine Situation s_0 (mit vollständiger Spezifikation)

Gesucht

- Eine Folge $\langle a_1, \ldots, a_n \rangle$ von primitiven Aktionen, so dass
- $\mathcal{KB} \models G(do(\langle a_1, \ldots, a_n \rangle, s_0)) \wedge Legal(\langle a_1, \ldots, a_n \rangle, s_0))$

Verfahren

- KI-Suchverfahren (Bei Bewertungsmöglichkeit: A*)
- (Resolutions-)Beweis mit Antwortextraktion
- Goal-Regression (Reiter 1991)

Aufzugs-Planung im Situationskalkül

Gegeben

- Ziel: Alle Anforderungen erfüllt, Parken im Erdgeschoss mit offener Tür.
- $\neg\exists n\ [on(n, s)] \wedge currentFloor(0, s) \wedge \neg d\_closed(s)$
- Startsituation s_0

Gesucht

- Eine Folge $\langle a_1, \ldots, a_n \rangle$ von primitiven Aktionen, so dass
- $\mathcal{KB} \models G(do(\langle a_1, \ldots, a_n \rangle, s_0)) \wedge Legal(\langle a_1, \ldots, a_n \rangle, s_0))$

Lösung

⟨close, down(3), open, turnoff(3), close, up(5), open, turnoff(5), close, down(0), open⟩

$s = do(open, do(down(0), do(close, do(turnoff(5), do(open, do(up(5), do(close, do(turnoff(3), do(open, do(down(3), do(close, s_0)))))))))))$

Aufzugsbeispiel: Mögliche Aktionen

		cFl	on	d_cl	Poss(a)
s0		4	3, 5	f	close
s1	do(close, s0)	4	3, 5	t	open, down(0)…(3), up(5)
s2	do(down(3), s1)	3	3, 5	t	open, down(0)…(2), up(4)…(5)
s3	do(open, s2)	3	3, 5	f	turnoff, close
s4	do(turnoff(3), s3)	3	5	f	close
s5	do(close, s4)	3	5	t	open, down(0)…(2), up(4)…(5)
s6	do(up(5), s5)	5	5	t	open, down(0)…(4)
s7	do(open, s6)	5	5	f	turnoff, close
s8	do(turnoff(5), s7)	5		f	close
s9	do(close, s8)	5		t	open, down(0)…(4)
s10	do(down(0), s9)	0		t	open, up(1)…(5)
s11	do(open, s10)	0		f	close

Beschränkungen des Situationskalküls (1)

Ereignisse als partielle Faktenwechsel

- Ereignisse sind ausschließlich als Situations- bzw. Faktenwechsel integrierbar.
  - *ein Knall oder die zweimalige Umrundung eines Sportplatzes durch einen Läufer*
  - Wie unterscheiden sich die vorhergehenden und nachfolgenden Situationen?
    Das Ereignis hat noch nicht / hat schon stattgefunden

Beschränkungen des Situationskalküls (2)

Komplexe Handlungen, Prozesse

- nicht erweiterbar auf simultan stattfindende distinkte Handlungen
  Er fuhr in die Stadt und hörte dabei Radio.
- Ursache und Wirkung müssen unmittelbar aufeinander folgen.
  - Prozessergebnisse: *Er lernte stetig. Ein Jahr später schaffte er die Prüfung mühelos*

Temporal/kausale Ordnung der Situationen ist im Situationskalkül diskret

- keine Repräsentation kontinuierlicher Veränderung
  Die Badewanne wird mit Wasser gefüllt

Literatur

McCarthy, J. & P.J. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer & D. Michie (eds.), *Machine Intelligence Vol. 4* (pp. 463-502). Edinburgh: Edinburgh University Press.

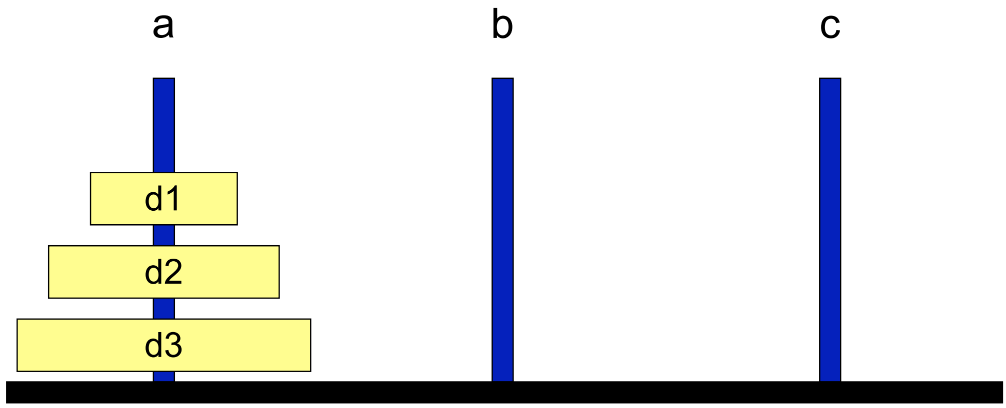
Levesque, Hector J., Raymond Reiter, Yves Lespérance, Fangzhen Lin & Richard B. Scherl (1997). GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming 31*. 59–84.

Brachman, Ronald J. & Hector J. Levesque (2004). *Knowledge Representation and Reasoning*. Morgan Kaufman. Chapter 14.

Reiter, Raymond (1991). The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz (ed.) *Artificial Intelligence and Mathematical Theory of Computation* (pp. 359–380). Academic Press: Boston.

Aktionen für den Turm von Hanoi

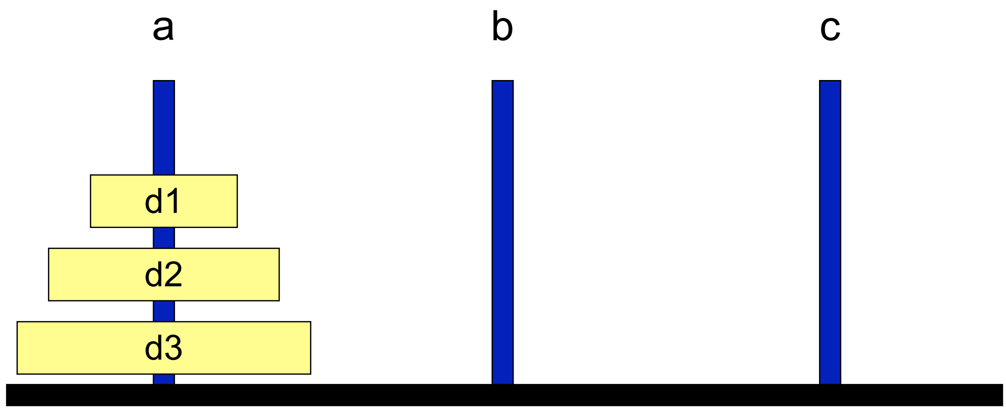
Gruppen-
diskussion



Welche Relationen, Fluents, Aktionen werden benötigt?
Welche Vorbedingungen und Effekte haben die Aktionen?

Aktionen für den Turm von Hanoi

Gruppen-
diskussion



Welche Axiome des Nachfolgezustands gelten für die Fluents?