

Wissensrepräsentation

Christopher Habel, Özgür Özçep
Sommersemester 2005

Sitzung 4: Wissensrepräsentation und Agenten

- (Wissens-)Repräsentation
- Wissensbasierte Agenten
- Struktur von Wissensbasen
- Regeltypen (Regeln: Logik und Produktionssysteme)

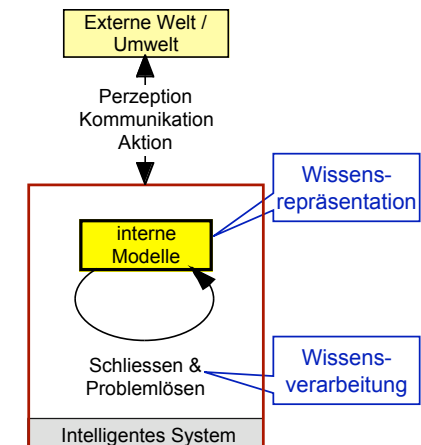
Wissensrepräsentation & -verarbeitung

Berechnung der Lösung

- Algorithmus

Suche / Herleitung der Lösung

- Problem- / Lösungsraum
- Suchstrategien
- Bewertungen / Heuristiken
- Schlussverfahren



Wissensrepräsentation, SoSe 2005
Ch. Habel / C. Eschenbach

4 - 2

Allgemeine Repräsentationstheorie: Palmer (1978)

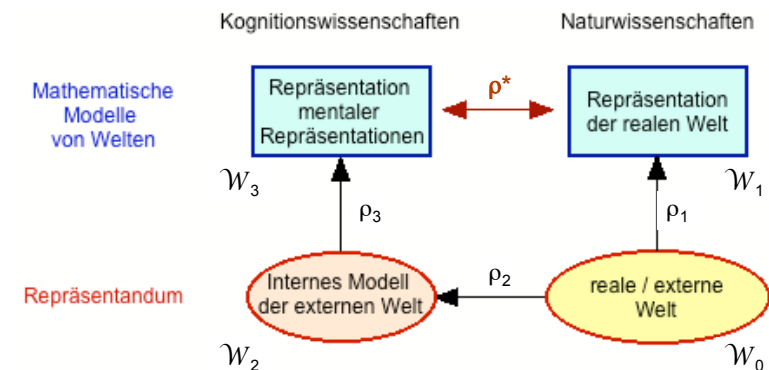
Repräsentationssituation

- repräsentierte Welt (Repräsentandum) \mathcal{W}_1
 - Aspekte, die modelliert werden (Struktur)
- repräsentierende Welt (Repräsentation) \mathcal{W}_2
 - Aspekte, die modellieren (Struktur)
- Korrespondenz (Repräsentationsabbildung) ρ
 - $\rho: \mathcal{W}_1 \rightarrow \mathcal{W}_2$

Keine Isomorphie

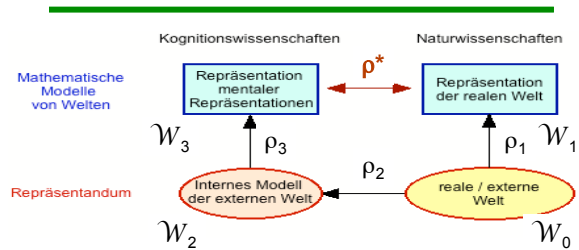
- (Un-) Vollständigkeit
- Detailliertheit, Genauigkeit, Exaktheit, ...

Formale Modelle in den Wissenschaften



Habel (1994, 2002).

Formale Modelle in den Wissenschaften



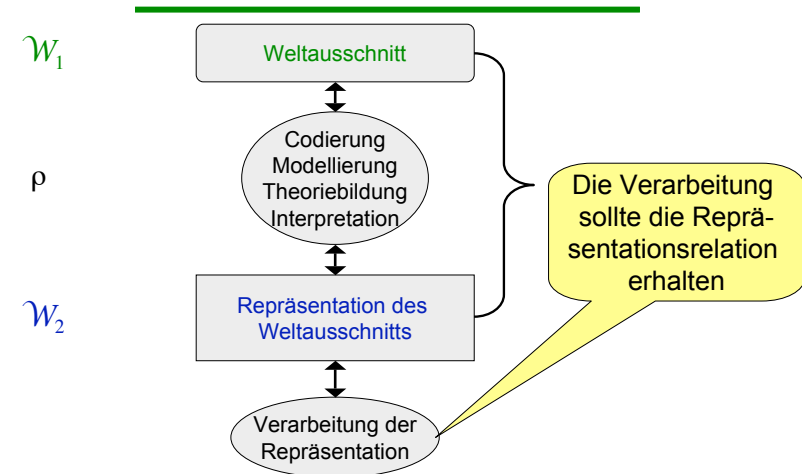
Fokus auf W_1 :

Ergänzung / Unterstützung der menschlichen Leistung
Informationssysteme, Expertensysteme, Planungshilfen

Fokus auf W_3 :

Nachvollziehen der menschlichen Leistung
Kommunikation, Schnittstellen, Robotik, Kognitionswissenschaft

(Wissens-)Repräsentation und Verarbeitung



Repräsentationen beim Problemlösen (1)

KI-Forschungsstrategie

- Trennung in allgemeine Problemlösungsverfahren und symbolische Repräsentationen der speziellen Problemstellung
- Eine adäquate Repräsentation ist essentiell für die Leistungsfähigkeit des Problemlösungsverfahrens

Repräsentationen beim Problemlösen (2)

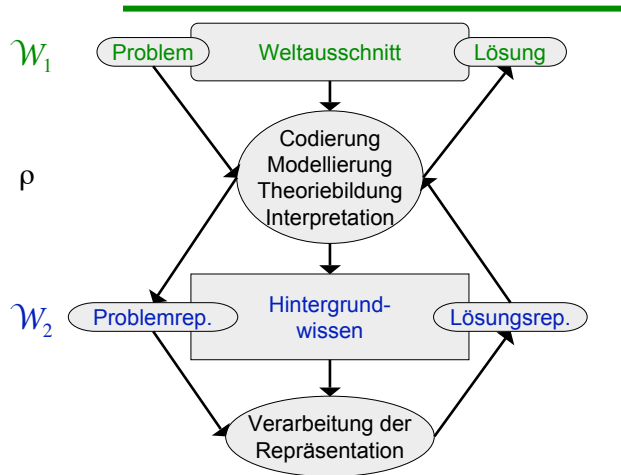
Prozedurale Darstellung / spezifischer Algorithmus

- Festlegung des Lösungswegs
- implizit kodiertes Wissen

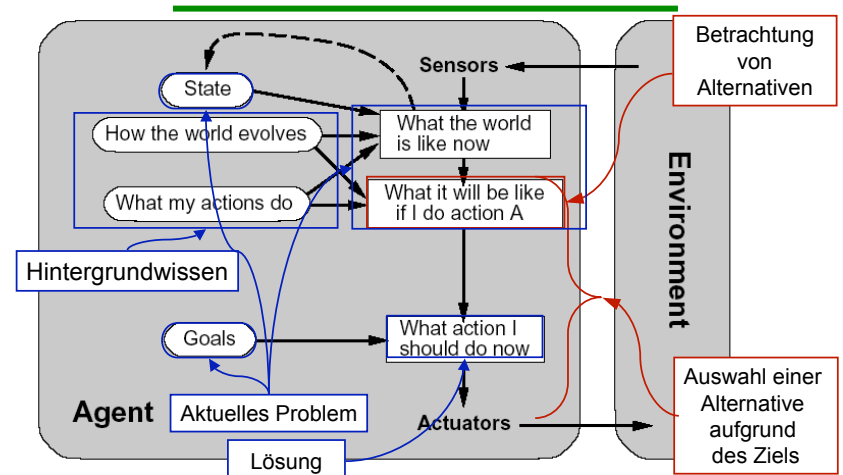
Deklarative Darstellung

- Trennung von Wissensbasis und Verarbeitung
- explizit kodiertes Wissen
 - Hintergrundwissen
 - Beziehung zwischen Elementen der Repräsentation, bzw. der Repräsentationssprache
 - Repräsentation des aktuell zu lösenden Problems
 - Wissen über die aktuelle Situation / Weltausschnitt
 - Ziel
- erschließbares Wissen (implizites Wissen)

Wissensbasierte Problemlösung



Goal-based agent



Wissensbasierte Agenten (knowledge-based agents)

function KB-AGENT(*percept*) returns an *action*

static: *KB*, a knowledge base

t, a counter, initially 0, indicating time

TELL(*KB*, MAKE-PERCEPT-SENTENCE(*percept*, *t*))

action ← ASK(*KB*, MAKE-ACTION-QUERY(*t*))

TELL(*KB*, MAKE-ACTION-SENTENCE(*action*, *t*))

t ← *t* + 1

return *action*

MAKE-PERCEPT-SENTENCE

→ Repräsentation, dass der Agent zur Zeit *t* *percept* wahrnimmt.

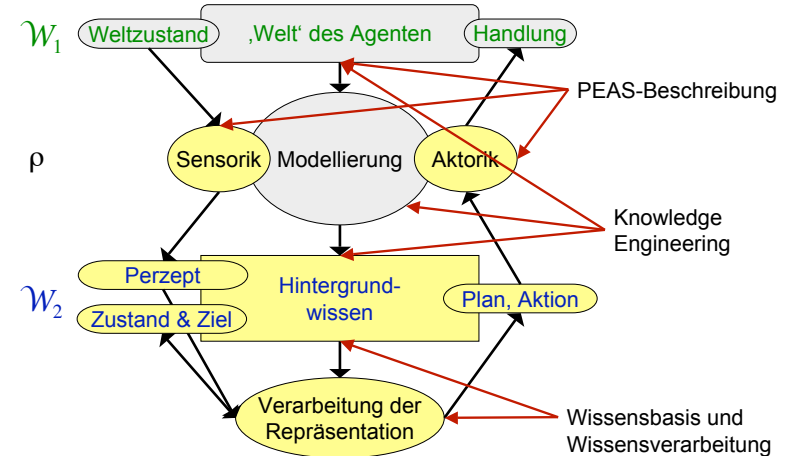
MAKE-ACTION-QUERY

→ Frage nach geeigneter Aktion zu *t*

MAKE-ACTION-SENTENCE

→ Repräsentation, dass der Agent zur Zeit *t* *action* ausführen will.

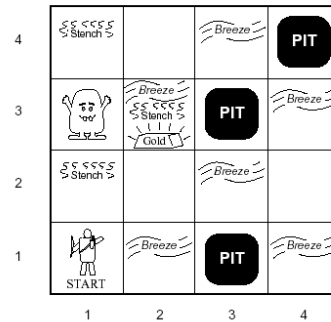
Wissensbasierter Agent



WUMPUS World

Eine **Höhlenwelt**, bestehend aus:

- Ein Agent, der Gold sucht. Er besitzt einen Bogen mit einem Pfeil.
- Ein Monster (Wumpus), das jeden frisst, der seine Kammer betritt.
- Fallgruben, in die der Agent fallen kann, aber nicht der Wumpus.



Eine WUMPUS
Start-Konfiguration

Wissen für die Wumpus-Welt: Eine quasi-formale Spezifikation

- Jede Zelle hat höchstens einen W-Nachbar.
- Wenn Zelle x W-Nachbar von Zelle y ist, dann ist y O-Nachbar von x.
- Zelle y ist genau dann Nachbar von Zelle x, wenn y W-Nachbar, O-Nachbar, N-Nachbar oder S-Nachbar von x ist.
- In einer Zelle gibt es genau dann einen Luftzug, wenn in (mind.) einer Nachbarzelle eine Fallgrube ist.
- Der Agent perzipiert genau dann einen Luftzug, wenn er sich in einer Zelle mit Luftzug befindet.
- Wenn A in Zelle x ist, nach W ausgerichtet ist, x einen W-Nachbar hat und A vorwärts geht, dann ist A anschließend in der W-Nachbarzelle von x.

Wissen für die Wumpus-Welt: Spezifikation in Prädikatenlogik (1)

- Jede Zelle hat höchstens einen W-Nachbar.
 $\forall x, y, z [W\text{-Nachbar}(y, x) \wedge W\text{-Nachbar}(z, x) \Rightarrow y = z]$
- Wenn Zelle x W-Nachbar von Zelle y ist, dann ist y O-Nachbar von x.
 $\forall x, y [W\text{-Nachbar}(x, y) \Rightarrow O\text{-Nachbar}(y, x)]$
- Zelle y ist genau dann Nachbar von Zelle x, wenn y W-Nachbar, O-Nachbar, N-Nachbar oder S-Nachbar von x ist.
 $\forall x, y [Nachbar(y, x) \Leftrightarrow (W\text{-Nachbar}(y, x) \vee O\text{-Nachbar}(y, x) \vee N\text{-Nachbar}(y, x) \vee S\text{-Nachbar}(y, x))]$
- In einer Zelle gibt es genau dann einen Luftzug, wenn in (mind.) einer Nachbarzelle eine Fallgrube ist.
 $\forall x [Luftzug(x) \Leftrightarrow \exists y [Nachbar(y, x) \wedge hatFallgrube(y)]]$

Wissen für die Wumpus-Welt: Spezifikation in Prädikatenlogik (2)

- Der Agent perzipiert genau dann einen Luftzug, wenn er sich in einer Zelle mit Luftzug befindet.
 - Ort des Agenten ist zeitabhängig (Agent ist mobil)
 - Wahrnehmung ist orts- und damit zeitabhängig.
 $\forall t [spürtLuftzug(A, t) \Leftrightarrow \exists x [in(A, x, t) \wedge Luftzug(x)]]$
- Wenn A in Zelle x ist, nach W ausgerichtet ist, x einen W-Nachbar hat und A vorwärts geht, dann ist A anschließend in der W-Nachbarzelle von x.
 - Aktionen sind nicht wahr oder falsch sondern werden ausgeführt oder nicht
 - Aktionen bewirken Veränderungen
 $\forall t, x, y [in(A, x, t) \wedge W\text{-Ausrichtung}(A, t) \wedge W\text{-Nachbar}(y, x) \wedge Do(t, A, forward) \Rightarrow in(A, y, next(t))]$

Wissen für die Wumpus-Welt: Spezifikation von Anfang und Ziel

- Die Startposition des Agenten ist [1, 1].
- Am Anfang hat der Agent einen Pfeil und kein Gold.
- Am Anfang lebt der Agent.
- Am Anfang befindet sich (irgendwo) in der Höhle Gold und ein lebender Wumpus.

- Ziel des Agenten ist, mit dem Gold (lebend) die Höhle zu verlassen.
- Nur im Notfall sollte von Waffen Gebrauch gemacht werden.

Präsenzaufgabe: Schlagen Sie Formalisierungen vor.
Erläutern Sie Besonderheiten und Herausforderungen

Wissen für die Wumpus-Welt: Weitere Regeln

- Wenn du dich an demselben Ort befindest wie das Gold und das Gold nicht hast, dann nimm das Gold.
- Wenn du das Gold siehst, dann bist Du an demselben Ort wie das Gold und hast es nicht.
- Wenn Du das Gold hast, dann verlasse die Höhle auf dem kürzesten sicheren Weg, den Du kennst.
- Gehe nur in Zellen, die sicher sind.
- Wenn eine Zelle den lebenden Wumpus enthält, dann ist sie nicht sicher.
- Wenn eine Zelle sicher ist, dann enthält sie weder einen lebenden Wumpus noch eine Fallgrube.

Präsenzaufgabe: Schlagen Sie Formalisierungen vor.
Erläutern Sie Besonderheiten und Herausforderungen

Arten von Wissen(-sbasen) in der Wumpus-Welt

Generelle Weltbeschreibung

- Räumliche Struktur der Welt
- Zeitlich: nur der Agent kann sich bewegen (?)
- Handlungsbezogen: Restriktionen und Effekte

Aktuelle Problem- und Lösungsbeschreibung

- gegenwärtige Position des Agenten
- gegenwärtige Perzepte
- im aktuellen Durchlauf gesammelte Information über die spezifische Situation / Konfiguration
- (aktuelles) Ziel, Plan

Erfahrungen

- ‚frühere Fälle‘

Struktur von Wissensbasen

Korrespondenz: Wissensbasen – Gedächtnis

generell: semantisch / konzeptuell

- Beziehungen zwischen Repräsentationseinheiten

aktuell: Arbeitsgedächtnis

- aktuelle Aufgaben, Umwelt, Pläne

Erfahrungen: episodisch

- Individuelle Aufgaben, Konstellationen, Lösungen

Struktur der Wissensbasis (2)

Explizites Wissen

- Inhalt der Wissensbasis
- Fakten und Regeln (\approx Regularitäten)

+ Schluss- / Verarbeitungsmethoden

- operieren auf explizitem Wissen
- Ableitungs-, Beweisverfahren

→ implizites Wissen

- ableitbar aus der Wissensbasis
- Korrekte Schlüsse (→ Wahrheit)
 - Folgerbar (→ Logik)
- Kombination von Fakten und Regeln
 - resultiert meistens in neuen Fakten

Regeltypen: Wenn A, dann B.

Rein Repräsentations- / Wahrheitsbezogen

- Wenn A (wahr ist), dann (ist) B (wahr).
- Wenn (du an) A (glaubst), dann (mußt du) B (glauben).
- Logik

Aktionsbezogen: Dynamic view of rules (B&L)

- Wenn A (wahr ist), dann (führe) B (aus).
- Wenn (du) A (glaubst), dann (führe) B (aus).
- Handlungskontrolle / Planung

Zielbezogen

- Wenn (du) A (erreichst), dann (erreichst du auch) B.
- Handlungskontrolle / Planung

Regeltypen (2)

Produktionsregeln: Aktionen auf der Wissensbasis

- Wenn A (wahr ist), dann füge B in die Wissensbasis ein.
- Wenn (du) A (ausführst), dann füge B in die Wissensbasis ein.

Negation ?

- Wenn A (wahr ist), dann lösche C aus der Wissensbasis.
- Wenn (du) A (ausführst), dann lösche C aus der Wissensbasis

➤ Beispiele: STRIPS, OPS5

Logik als Repräsentationssprache

Lange Tradition

- Prinzipien der denotationellen Semantik
- laborierte Ableitungsmethoden
- Grenzen der Berechenbarkeit

Semantik

- Wahrheitsbezug:
 - Zentrale syntaktische Kategorien sind Propositionen / Aussagen
- Kompositionalität der Wahrheitswertberechnung
- Folgerung
- Äquivalenz

Wissensbasen und Folgerung

Entailment means that one thing *follows from* another:

$$KB \models \alpha$$

Knowledge base KB entails sentence α
if and only if
 α is true in all worlds where KB is true

- Wenn KB die Welt korrekt beschreibt
- und $KB \models \alpha$,
- dann beschreiben auch α und $KB \cup \{\alpha\}$ die Welt korrekt.
- Ergänzung von α zerstört bestehende Rep.-Beziehungen nicht

Aus der Perspektive eines Agenten

- Wenn $KB \models \alpha$,
- dann ist der Agent mit KB berechtigt, auf α zu schließen.

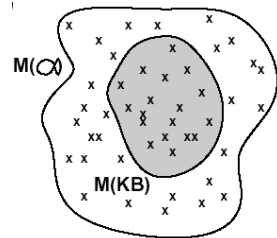
Wissensbasen: Folgerung & Modelle

Logicians typically think in terms of **models**, which are formally structured worlds with respect to which truth can be evaluated

We say m is a **model** of a sentence α if α is true in m

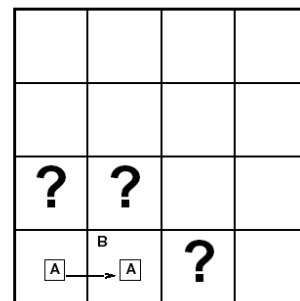
$M(\alpha)$ is the set of all models of α

Then $KB \models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$



Folgerung in der WUMPUS-Welt

Situation after detecting nothing in $[1,1]$,
moving right, breeze in $[2,1]$

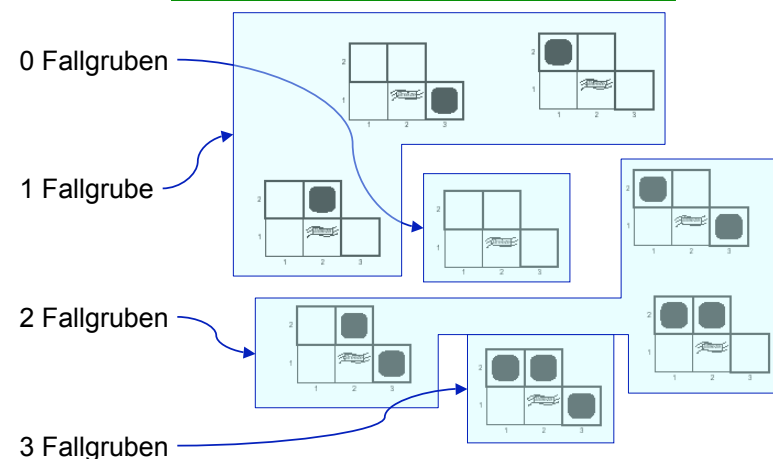


Consider possible models for ?s
assuming only pits

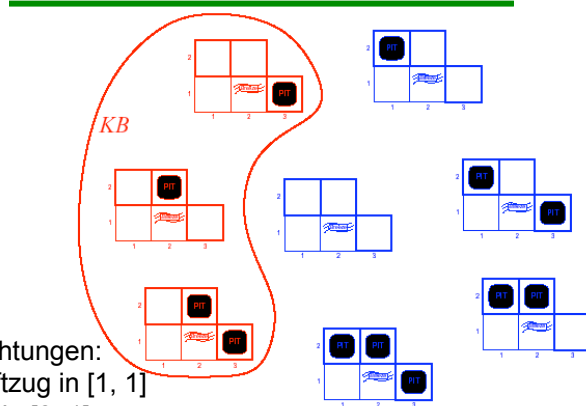
3 Boolean choices \Rightarrow 8 possible models

Für 3 Kammern jeweils + / - Fallgrube.

Mögliche Verteilung von Fallgruben auf drei Kammern



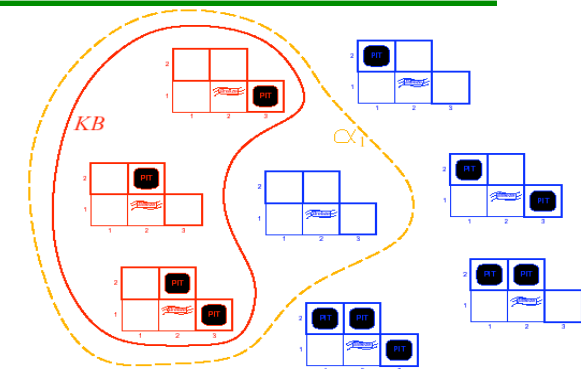
Modelle und Inhalt der Wissensbasis des Agenten



Beobachtungen:
kein Luftzug in [1, 1]
Luftzug in [2, 1]

$KB = \text{wumpus-world rules} + \text{observations}$

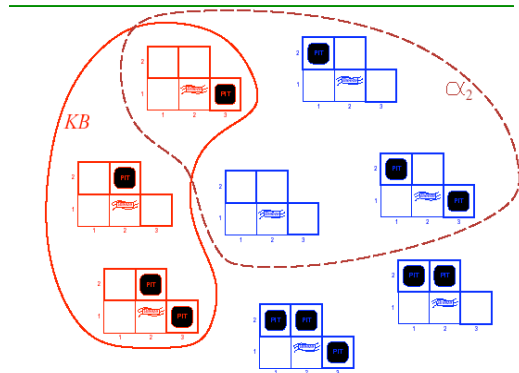
Beweis durch ‚Model-checking‘



$KB = \text{wumpus-world rules} + \text{observations}$

$\alpha_1 = \text{“[1,2] is safe”}$; $KB \models \alpha_1$, proved by model checking

Gegenbeweis durch ‚Model-checking‘



$KB = \text{wumpus-world rules} + \text{observations}$

$\alpha_2 = \text{“[2,2] is safe”}$, $KB \not\models \alpha_2$

2.1 Deductive Reasoning Agents

- How can an agent decide what to do using theorem proving?
- Basic idea is to use logic to encode a theory stating the *best* action to perform in any given situation.
- Let:
 - ρ be this theory (typically a set of rules);
 - Δ be a logical database that describes the current state of the world;
 - Ac be the set of actions the agent can perform;
 - $\Delta \vdash_{\rho} \phi$ mean that ϕ can be proved from Δ using ρ .

```

/* try to find an action explicitly prescribed */
for each  $a \in Ac$  do
  if  $\Delta \vdash_{\rho} Do(a)$  then
    return  $a$ 
  end-if
end-for
/* try to find an action not excluded */
for each  $a \in Ac$  do
  if  $\Delta \not\vdash_{\rho} \neg Do(a)$  then
    return  $a$ 
  end-if
end-for
return null /* no action found */

```

Entscheidbarkeit ?

Was, wenn $Do(a)$ und $Do(b)$
($a \neq b$) ableitbar sind ?

Was bedeutet $Do(a)$?

<http://www.csc.liv.ac.uk/~njw/pubs/imas/>

8

Aktionsauswahl und Logik

Probleme

- Prädikatenlogik erster Stufe ist semi-entscheidbar
- Selbst in der Aussagenlogik kann ein Beweis sehr aufwendig sein
- Beweisverfahren können die Struktur der Welt nicht nutzen

Lösungen

- Aufwandsbeschränkungen bei den Beweisen
- Einschränkung der Sprache
 - (Teilmenge der Prädikatenlogischen Sprache)
- spezifischere Verfahren, die die Struktur des Problemraums nutzen können

Vorschau: Produktionssysteme

Grundprinzip

- Arbeitsspeicher: Fakten
 - Problem, partielle Lösung, ...
- Hintergrundwissen: Bedingungs-Aktions-Regeln (Produktionen)
 - Linke Seite: LHS: Bedingungen
 - Rechte Seite: RHS: Aktionen

Recognize-Act-Zyklus

- 1) Stelle fest, welche LHS durch den aktuellen Zustand des Arbeitsspeichers erfüllt wird.
- 2) Wähle eine der zugehörigen Produktionen aus.
- 3) Führe die Aktionen der entsprechenden RHS aus.
- 4) Weiter bei 1)