

# Wissensrepräsentation

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

Sommersemester 2003

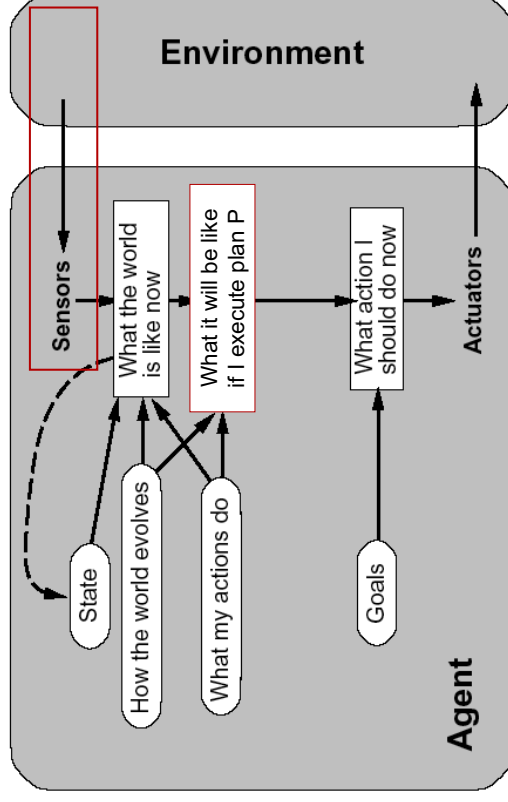
# Wissensrepräsentation

Christopher Habel, Hedda Schmidtke  
Sommersemester 2003

## Sitzung 27: Aktionen und Wahrnehmung

- Erweiterung Situationskalkül
- Wissensoperator
- Wahrnehmungsoperatoren:  
Wissenserzeugende Aktionen

## Planning Goal-based agent



## Wahrnehmung im Situationskalkül

### 'Nur' notwendig wenn

- der Agent nicht alles weiß bzw. vorhersagen kann

### Epistemische Fluents

- $\text{Knows}(\Theta, s)$ : Der Agent weiß in Situation  $s$ , dass  $\Theta$  wahr ist.
- $\text{Knows}(\neg\Theta, s)$ : Der Agent weiß in Situation  $s$ , dass  $\Theta$  falsch ist.
- $\neg\text{Knows}(\Theta, s)$ : Es ist nicht der Fall, dass der Agent in Situation  $s$  weiß, ob  $\Theta$  wahr ist.

### An Modallogik orientierter Ansatz

- Unwissen wird durch eine Alternativmenge von Situationen modelliert, die sich in den Wahrheitswerten der unbekanntem Aussagen unterscheiden.
- Der Agent weiß nicht, in welcher Situation er sich befindet.

Wissensrepräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtko

27 - 5

## Wissenstypen

$\text{Knows}(\Theta, s) \Leftrightarrow \forall s' [K(s', s) \Rightarrow \Theta(s')]$

### Wissen, ob $\Theta$ wahr oder falsch ist

- $\text{Kwheather}(\Theta, s) \Leftrightarrow \text{Knows}(\Theta, s) \vee \text{Knows}(\neg\Theta, s)$

### Wissen, wen oder was $\tau$ bezeichnet

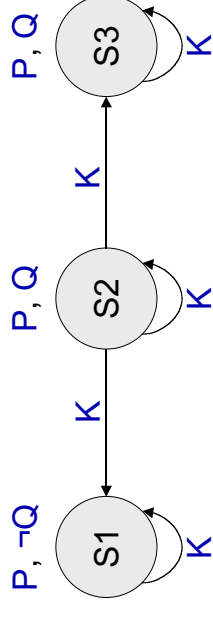
- $\text{Kref}(\tau, s) \Leftrightarrow \exists x [\text{Knows}(\tau = x, s)]$ , wobei  $x$  in  $\tau$  nicht frei vorkommt
- NB: Ist dieser Ausdruck tatsächlich angemessen, um Kenntnis von Referenz auf die Welt auszudrücken ?
- Alternative (?)  $\text{Kref}(\tau, s) \Leftrightarrow \exists a [\text{Knows}(\tau = \text{SR}(a), s)]$  mit Zusatzbedingung, dass  $a$  in  $s$  auftritt ?

Wissensrepräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtko

27 - 7

## Beispiel für Unkenntnis eines Agenten



### Interpretation von $K(s', s)$

- In der Situation  $s$  hält der Agent es für möglich, dass er sich in Situation  $s'$  befindet.

### Wissen = Notwendig bzgl. alternativen Situationen

$\text{Knows}(\Theta, s) \Leftrightarrow \forall s' [K(s', s) \Rightarrow \Theta(s')]$

Agent in  $S_2$  weiß, dass  $P$  wahr ist, aber nicht, ob  $Q$  wahr ist.

### Randbedingung

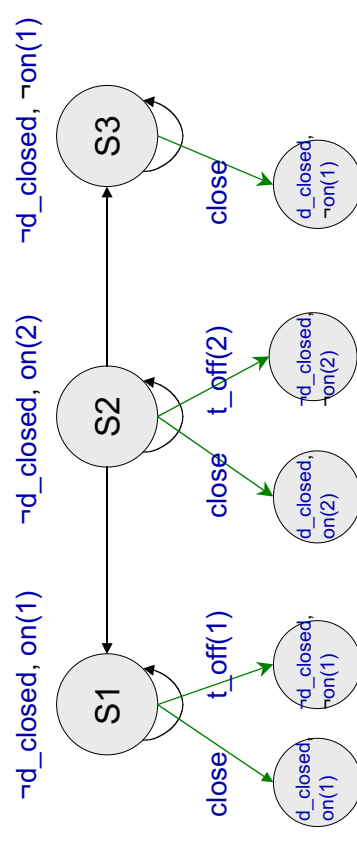
$K$  ist reflexiv (Theorie des Wissens)

Wissensrepräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtko

27 - 6

## Beispiel für Fortschreibung von Wissen



$\text{Knows}(\text{d\_closed}(\text{do}(\text{close}, s)), s)$

### Wie ist zu erreichen

$\text{Knows}(\text{d\_closed}(\text{do}(\text{close}, s)), \text{do}(\text{close}, s)) ?$

Wissensrepräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtko

27 - 8

## Axiomatisierungspolitik: Zwei Typen von Aktionen

### 'Wissensproduzierende' Aktionen

- Haben auf den Fluent  $K(s', s)$  einen Effekt
- Haben auf keinen anderen Fluent Effekt
- Wahrnehmen, Lesen, ...

### andere (Standard-)Aktionen

- Haben auf den Fluent  $K(s', s)$  keinen (individuellen) Effekt
- Anforderung ausschalten, Tür schließen, Tür aufmachen, ...

### Kombination als Sequenz

- Tür aufmachen und 'raus schauen
- 'Fehlschlagen' von Aktionen nicht als Rückkopplung

## (Wissen im) Nachfolgezustand von Standardaktionen

### Wissen (Knows)

- wird implizit über die Relation  $K$  modelliert
- Welche Situation steht in Relation  $K$  zu  $do(a, s)$  ?
- Für alle **Standardaktionen** soll gelten  
 $K(s'', do(a, s)) \Leftrightarrow \exists s' [K(s', s) \wedge Poss(a, s') \wedge s'' = do(a, s')]$
- Bestimmung allein durch
  - $K$ -Einbettung von  $s$  und
  - Aktion  $a$

## Axiomatisierungspolitik: Zwei Typen von Aktionen

### 'Wissensproduzierende' Aktionen

- Haben auf den Fluent  $K(s', s)$  einen Effekt
- Haben auf keinen anderen Fluent Effekt
- Wahrnehmen, Lesen, ...

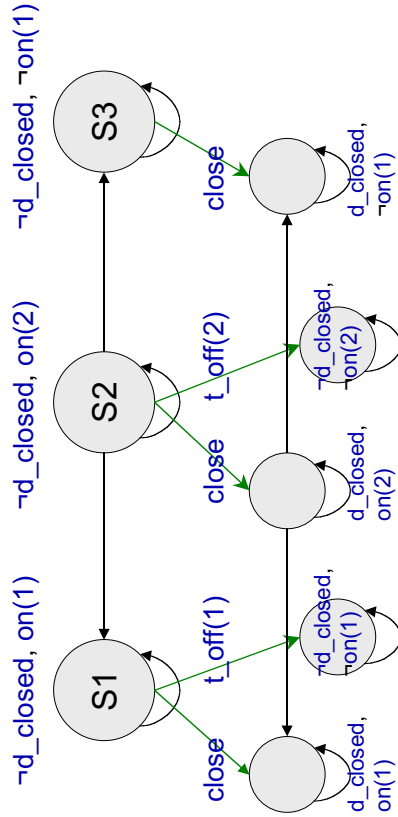
### andere (Standard-)Aktionen

- Haben auf den Fluent  $K(s', s)$  keinen (individuellen) Effekt
- Anforderung ausschalten, Tür schließen, Tür aufmachen, ...

### Kombination als Sequenz

- Tür aufmachen und 'raus schauen
- 'Fehlschlagen' von Aktionen nicht als Rückkopplung

## Beispiel für Fortschreibung von Wissen



$Knows(d\_closed(do(close, s)), do(close, s))$

## Sensor-Aktionen: Wissenserwerb

### Sensor-Aktionen: Beispiel

- (Lokale) Inspektion der Welt:  $Q$ -Sensor-Aktion:  $Sense_Q$
- Liefert Information über den Fluent  $Q$  in Situation  $s$ 
  - Sensor-Resultat  $r$  (ist verlässlich)
  - Fluent  $SR$
  - $SR(Sense_Q, s) = r \Leftrightarrow (r = "yes" \wedge Q(s)) \vee (r = "no" \wedge \neg Q(s))$
  - haben keinen Effekt auf den Wahrheitswert von  $Q$ 
    - $Q(do(Sense_Q, s)) \Leftrightarrow Q(s)$
  - haben Vorbedingungen:  $Poss(Sense_Q, s) \Leftrightarrow \dots$
  - haben (nur) auf den Wissenszustand Effekte
    - $K(s'', do(Sense_Q, s)) \Leftrightarrow \exists s' [K(s', s) \wedge Poss(Sense_Q, s') \wedge s'' = do(Sense_Q, s') \wedge SR(Sense_Q, s) = SR(Sense_Q, s')]$

## Sense<sub>Q</sub>: Effekt auf den Wissenszustand

- Wenn  $SR(Sense_Q, s) = \text{"yes"}$   
 $SR(Sense_Q, s) = r \Leftrightarrow (r = \text{"yes"} \wedge Q(s)) \vee (r = \text{"no"} \wedge \neg Q(s))$
- dann  $Q(s)$  und mit  
 $Q(do(Sense_Q, s)) \Leftrightarrow Q(s)$
- auch  $Q(do(Sense_Q, s))$
- und für alle  $s'$  mit  $SR(Sense_Q, s) = SR(Sense_Q, s')$  gilt  $Q(s')$   
und  $Q(do(Sense_Q, s'))$
- für alle  $s''$  mit  $K(s''), do(Sense_Q, s)$   
 $K(s'', do(Sense_Q, s)) \Leftrightarrow \exists s' [K(s', s) \wedge Poss(Sense_Q, s') \wedge$   
 $s'' = do(Sense_Q, s') \wedge SR(Sense_Q, s) = SR(Sense_Q, s')]$
- gilt auch  $Q(s'')$
- $Knows(Q, do(Sense_Q, s)) \Leftrightarrow$   
 $\forall s'' [K(s'', do(Sense_Q, s)) \Rightarrow Q(s'')]$
- also  $Knows(Q, do(Sense_Q, s))$

Wissenspräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtke

27 – 13

## Sense<sub>Q</sub>: Effekt auf den Wissenszustand

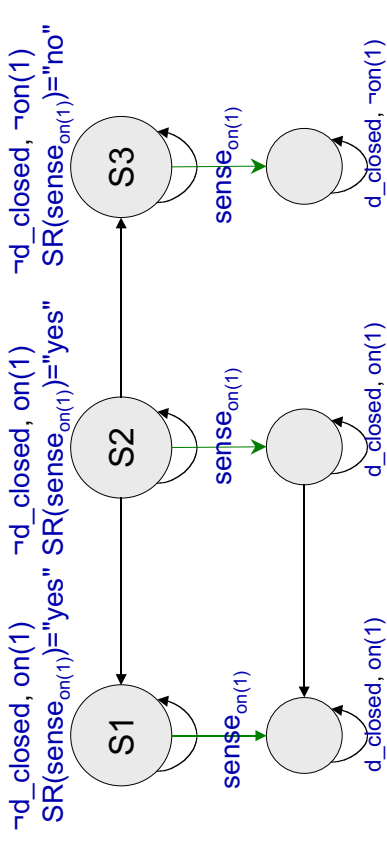
- Wenn  $SR(Sense_Q, s) = \text{"no"}$   
 $SR(Sense_Q, s) = r \Leftrightarrow (r = \text{"yes"} \wedge Q(s)) \vee (r = \text{"no"} \wedge \neg Q(s))$
- dann  $\neg Q(s)$  und mit  
 $Q(do(Sense_Q, s)) \Leftrightarrow Q(s)$
- auch  $\neg Q(do(Sense_Q, s))$
- und für alle  $s'$  mit  $SR(Sense_Q, s) = SR(Sense_Q, s')$  gilt  
 $\neg Q(s')$  und  $\neg Q(do(Sense_Q, s'))$
- für alle  $s''$  mit  $K(s''), do(Sense_Q, s)$   
 $K(s'', do(Sense_Q, s)) \Leftrightarrow \exists s' [K(s', s) \wedge Poss(Sense_Q, s') \wedge$   
 $s'' = do(Sense_Q, s') \wedge SR(Sense_Q, s) = SR(Sense_Q, s')]$
- gilt auch  $\neg Q(s'')$
- $Knows(\neg Q, do(Sense_Q, s)) \Leftrightarrow$   
 $\forall s'' [K(s'', do(Sense_Q, s)) \Rightarrow \neg Q(s'')]$
- also  $Knows(\neg Q, do(Sense_Q, s))$

Wissenspräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtke

27 – 15

## Beispiel für Kenntniserwerb eines Agenten



$K(s'', do(Sense_Q, s)) \Leftrightarrow \exists s' [K(s', s) \wedge Poss(Sense_Q, s') \wedge$   
 $s'' = do(Sense_Q, s') \wedge SR(Sense_Q, s) = SR(Sense_Q, s')]$   
 $Knows(on(1), do(sense_{on(1)}, S2))$

Wissenspräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtke

27 – 14

## Interaktion von SR und K

### Sensor-Resultat-Axiome

- Beispiel: Lesen  
 $SR(read(t), s) = r \Leftrightarrow r = content(t, s)$   
 $poss(read(t), s) \Rightarrow Kref(content(t), do(read(t), s))$
- Beispiel: Sense<sub>Q</sub>  
 $poss(Sense_Q, s) \Rightarrow Kwhether(Q, do(Sense_Q, s))$   
 $\forall s [Kwhether(\Theta, s) \Leftrightarrow Knows(\Theta, s) \vee Knows(\neg\Theta, s)]$
- Für Standardaktionen: Dummy-Resultat  
 $SR(a, s) = r \Leftrightarrow r = \text{"OK"}$

### Nachfolgezustandsaxiom für K

$\forall a, s, s'' [K(s'', do(a, s)) \Leftrightarrow \exists s' [K(s', s) \wedge s'' = do(a, s') \wedge$   
 $Poss(a, s') \wedge SR(a, s) = SR(a, s')]]$

Wissenspräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtke

27 – 16

## Beispiel: Safe öffnen

---

### Standardaktionen

$\text{poss}(\text{dial\_comb}(x), s) \Leftrightarrow \text{safe}(x) \wedge \text{at}(x, s) \wedge \text{Kref}(\text{comb}(x), s)$   
 $\text{SR}(\text{dial\_comb}(x), s) = r \Leftrightarrow r = \text{"OK"}$

### Wissensproduzierende Aktionen

$\text{poss}(\text{read}(x), s) \Leftrightarrow \text{at}(x, s)$   
 $\text{SR}(\text{read}(x), s) = r \Leftrightarrow r = \text{info}(x)$

### Nachfolgezustandsaxiome für Fluents

$\text{open}(x, \text{do}(a, s)) \Leftrightarrow$   
 $a = \text{dial\_comb}(x) \vee \text{open}(x, s) \wedge a \neq \text{lock}(x)$   
 $\text{at}(x, \text{do}(a, s)) \Leftrightarrow a = \text{move\_to}(x) \vee \text{at}(x, s) \wedge a \neq \text{move\_to}(y)$

Wissensrepräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtke

27 – 17

## Beispiel: Safe öffnen

---

### Hintergrundwissen

$\text{safe}(\text{SF})$

### Startsituation

$\text{Knows}(\text{info}(\text{PPR}) = \text{comb}(\text{SF}), s0)$   
 $\text{at}(\text{SF}, s0), \text{at}(\text{PPR}, s0)$

### Beweisbar

$\text{poss}(\text{read}(\text{PPR}), s0)$   
 $\text{SR}(\text{read}(\text{PPR}), s) = r \Leftrightarrow r = \text{info}(\text{PPR})$   
 $\text{Knows}(\text{info}(\text{PPR}) = \text{SR}(\text{read}(\text{PPR})), \text{do}(\text{read}(\text{PPR}), s0))$   
 $\text{Knows}(\text{comb}(\text{SF}) = \text{SR}(\text{read}(\text{PPR})), \text{do}(\text{read}(\text{PPR}), s0))$   
 $\text{Kref}(\text{comb}(\text{SF}), \text{do}(\text{read}(\text{PPR}), s0))$   
 $\text{poss}(\text{dial\_comb}(\text{SF}), \text{do}(\text{read}(\text{PPR}), s0))$

Wissensrepräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtke

27 – 18

## Generelle Eigenschaften des Ansatzes

---

### Wissenseffekte von Aktionen

- Wenn
  - der Agent weiß, dass in Situation  $s$  Fluent  $F$  wahr ist und Aktion  $a$  ausführbar ist
  - die Axiomatisierung besagt,
    - dass unter Voraussetzung  $F$  (in  $s$ )  $a$  dazu führt, dass Fluent  $P$  wahr ist
  - dann weiß der Agent nach Ausführung von  $a$  in  $s$ , dass  $P$  wahr ist

### Wissen über Aktionen

- Der Agent kennt die Spezifikation der Aktionen

Wissensrepräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtke

27 – 19

## Generelle Eigenschaften des Ansatzes

---

### Persistenz der Ignoranz

- Wenn
  - der Agent nicht weiß, ob Fluent  $P$  wahr ist
  - die Axiomatisierung besagt, dass die Aktion  $a$ 
    - $P$  unverändert lässt
    - keine Wahrnehmungssaktion ist, die  $P$  betrifft
  - dann weiß der Agent nach Ausführung von  $a$  auch nicht, ob  $P$  wahr ist

Wissensrepräsentation, SoSe 2004

Ch. Habel / C. Eschenbach / H. Schmidtke

27 – 20

## Generelle Eigenschaften des Ansatzes

### Wissenserwerb

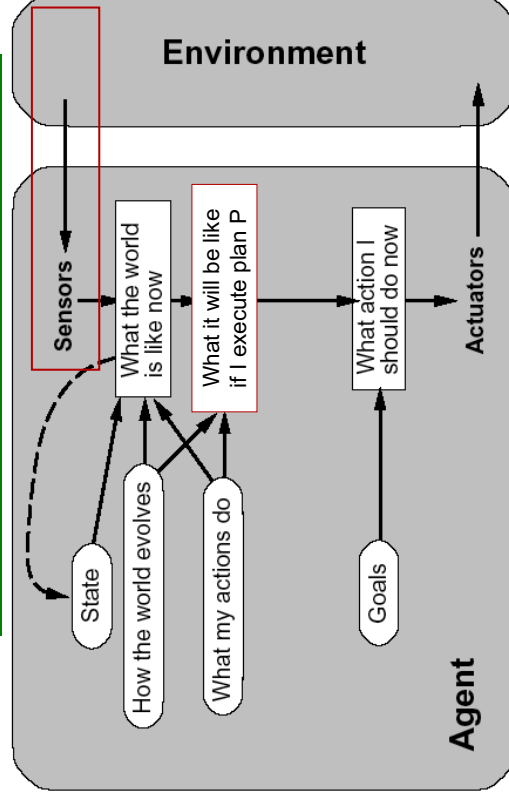
- Wenn
  - der Agent weiß,
    - dass ein Sensorresultat der Aktion **a** genau dann eintritt, wenn Fluent **F** wahr ist
    - dass **F** → **P** wahr ist
  - **F** in einer Situation **s** wahr ist und **a** ausführbar ist
- dann weiß der Agent nach Ausführung von **a** in **s** auch, dass **P** wahr ist

## Generelle Eigenschaften des Ansatzes

### Gedächtnis

- Wenn
  - der Agent weiß,
    - dass Fluent **P** in einer Situation **s** wahr ist
  - die Axiomatisierung besagt,
    - dass die Aktion **a** **P** nicht falsch macht
- dann weiß der Agent nach Ausführung von **a** in **s** auch, dass **P** wahr ist

## Planning Goal-based agent



## Diskussionsrunde

### Situationskalkül mit Wahrnehmungsaktionen

- erlaubt die Planung von Wissenserwerb
- Initialsituationen müssen nicht vollständig spezifiziert sein
- bietet aber noch keinen direkten Rahmen für den Handlungs-Wahrnehmungs-Zyklus von Agenten

### Was fehlt für den Aufbau einer Situationskalkül-basierten Agentenarchitektur?