

Artificial Intelligence und Wissensrepräsentation

Prof. Georg Gottlob

Institut für Informationssysteme

Technische Universität Wien

Folien zur Vorlesung “Konzepte der Artificial Intelligence”

Gliederung

- Zum Begriff “Artificial Intelligence”
- Zur (Vor-)Geschichte der AI
- Forschungsgebiete der AI
- Anwendungsgebiete der AI
- Überblick über die Vorlesung
- Wissensrepräsentation

Übersichtsvortrag, keine Details!

Zum Begriff “Artificial Intelligence”

AI: Interdisziplinäres Forschungsgebiet mit folgenden Hauptzielen:

- Simulation des menschlichen Gehirns, um seine Funktionsweise besser zu verstehen (Kognitionswissenschaften).
- Nachbildung menschlicher Problemlösungsfähigkeiten in Computerprogrammen (informatische AI).

Oft zu hohe Erwartungen führten zu Diskussionen und scharfen Entgegnungen von Philosophen und Wissenschaftskritikern.

Entgegnungen waren ihrerseits oft überzogen:

Hubert L. Dreyfus 1972 *What Computers Can't Do*: Computer werden nie einen Schachweltmeister schlagen.

1997: Deep Blue schlug Garey Kasparov.

Zur (Vor-)Geschichte der AI

Der Wunsch, menschliche Intelligenz mechanisch nachzubilden besteht seit vielen Jahrhunderten.

Er schloss den Wunsch ein, die Mathematik zu automatisieren.

Automatische Schachspieler. Erfindung verschiedener Maschinen, u.a.

Betrug durch versteckte menschliche Spieler.

Leibniz (1676-1711) *Calculus Ratiocinator* zur Automatisierung des Mathematischen Schließens

Hilbert (1862-1943) Programm zur vollständigen und konsistenten Axiomatisierung der Mathematik.

Russel und **Whitehead**: *Principia Mathematica* (1910) Versuch der Realisierung von Hilberts Programm.

Gödel 1931: *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme.*

Gödel zeigt, daß die Zahlentheorie nicht (rekursiv) axiomatisierbar ist.

Zu jeder Axiomatisierung A der Zahlentheorie gibt es einen wahren Satz G , sodass weder $A \vdash G$, noch $A \vdash \neg G$. (\vdash : ableitbar).

D.h. Jede (rekursive) Axiomatisierung der Zahlentheorie ist unvollständig.

Hiermit wurde Hilberts Programm zunichte gemacht. Der Traum einer “mechanisierbaren” Mathematik war zu Ende.

Church (1936): Die Prädikatenlogik ist unentscheidbar.

Seit diesen negativen Resultaten: Hoffnung, daß wenigstens Teilgebiete des Denkens, Entscheidens und Handelns automatisierbar sind.

Turing, 1950. Turing Test für maschinelle Intelligenz:

Ein nur über Fernschreiber mit je einem Menschen und einem Rechner verbundener Tester muß herausfinden, wer sein menschlicher und wer sein maschineller Partner ist. Gelingt das nicht, liegt künstliche Intelligenz vor.

(Seit ca. 1990: Jährlicher Wettbewerb mit Preis \$100.000.-, bis zum Bestehen des Tests).

Turing Test im Internet:

www.bbc.co.uk/education/hosc/diyexperiments/turing/turing_index.shtml

McCarthy, 1956: Führt den Begriff der *Artificial Intelligence* ein.

Postulate von McCarthy:

- Ein Intelligentes System muß eine *Repräsentation* der Welt besitzen, d.h. auf eine Menge von Fakten zugreifen, sie verarbeiten und erweitern können.
- Zur Repräsentation dienen Mengen von *logischen* Aussagen, bzw. Formeln, die Problemlösungen als Folgerungen enthalten.
- Die Problemlösung muß *auffindbar* sein. D.h., es muß eine wirksame Suchstrategie für Beweise existieren.

Methoden und Forschungsgebiete der AI

- Wissensrepräsentation
- Wissensverarbeitung, Schlußfolgern
- Unsicheres Schließen
- Automatisches Beweisen
- Heuristische Suche
- Planen
- Automatisches Lernen und Wissenserwerb



- Kognitives Modellieren (z.B. modellieren räumlicher Eindrücke)
- Qualitatives Modellieren und Schließen
- Automatische Diagnose
- Verstehen und Generieren Natürlicher Sprache
- Logisches Programmieren
- Neuronale Netze
- Intelligente Agenten

Anwendungsgebiete der AI

- Expertensysteme und Entscheidungsunterstützung
- Konfiguration von elektronischen Geräten und Anlagen
- Medizinische Diagnose
- Diagnose und Überprüfung technischer Anlagen
- Game playing, automatische Spieler
- Robotersteuerung

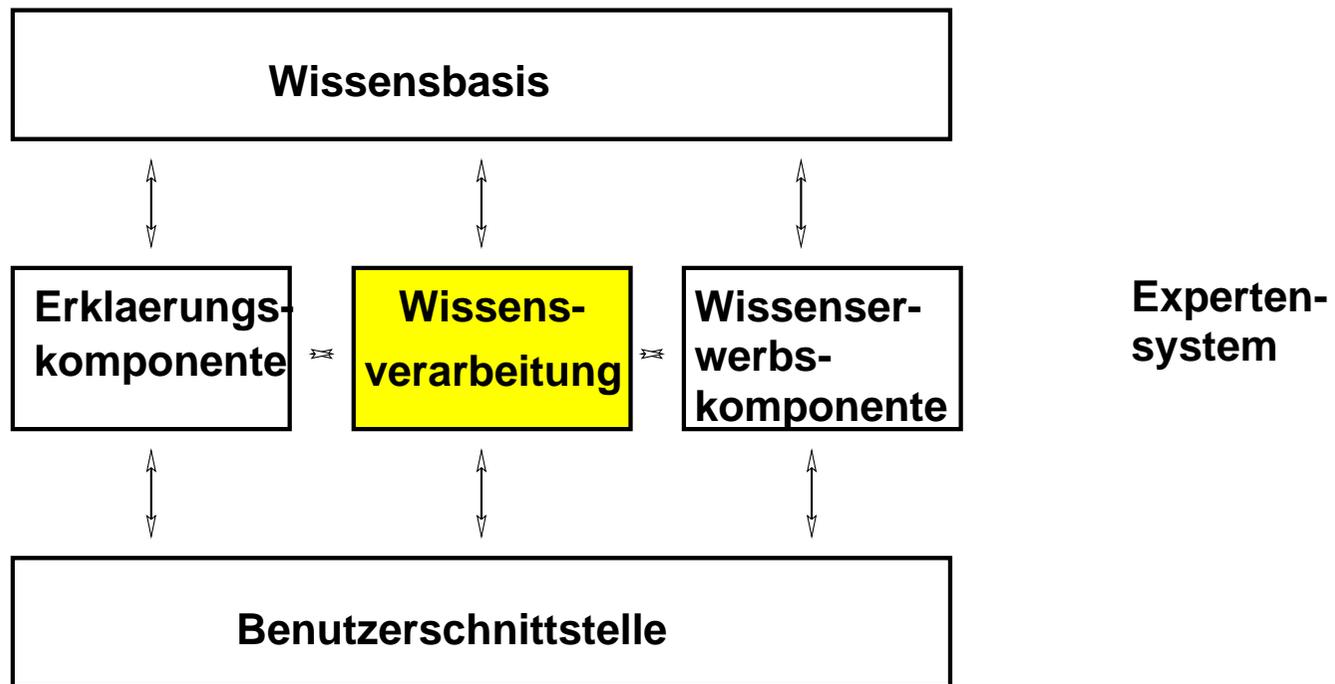


- Sprachübersetzungssysteme, natürlichsprachiger Datenbankzugriff
- Scheduling, Ablaufsteuerung und Optimierung industrieller Prozesse,
- Informationsgewinnung aus dem Internet
- Shopping agents, automatisches Verhandeln
- Computerunterstütztes Lernen
- Mustererkennung
- Bildverstehen

Wissensrepräsentation und -Verarbeitung

Wird detailliert in der VO *Wissensbasierte Systeme* behandelt.

- Prozedurale Wissensrepräsentation
- Objektorientierte Wissensrepräsentation
- Logische Wissensrepräsentation und -verarbeitung.



Prozedurale Wissensrepräsentation

Wissensbasis = Programm.

Z.B. für jedes Prädikat eine Prozedur.

```
Procedure vogel(x:name): boolean;  
    if x='Taube' then return true else  
        (if x = 'Kanarienvogel' then return true else  
            (if x = 'Geier' then return true else  
                (if x= 'Pinguin' then return true else  
                    (if ... .. else return false ...))))  
end.
```

```
Procedure kannfliegen(x:name):  boolean
    if flugzeug(x) then return true else
        (if schmetterling(x) then return true else
            (if pinguin(x) then return false else
                (if vogel(x) then return true else
                    (if ... .. else return false))))
end.
```

Vorteil: sehr effizient

Nachteil: aufwendig zu ändern, falls sich Wissen ändert.

Objektorientierte Wissensrepräsentation

1.) Frames

Frame-Idee von Marvin Minsky (1975):

When one encounters a new situation (or makes a substantial change in one's view of the present problem) one selects from memory a substantial structure called a frame. This is a remembered framework to be adapted to fit reality by changing details as necessary...

Konkretisierung der Frame-Idee

Frame: Menge von *slots* (attribute).

Spezielle slots: AKO (a kind of) und INSTANCE (verweist auf Instanzen)

Slots werden durch *facets* beschrieben. Facets können Werte, Listen von Werten, Bedingungen oder ausführbaren Code (in Lisp) enthalten.

Vererbung von Attributen und Werten spielt eine wichtige Rolle.

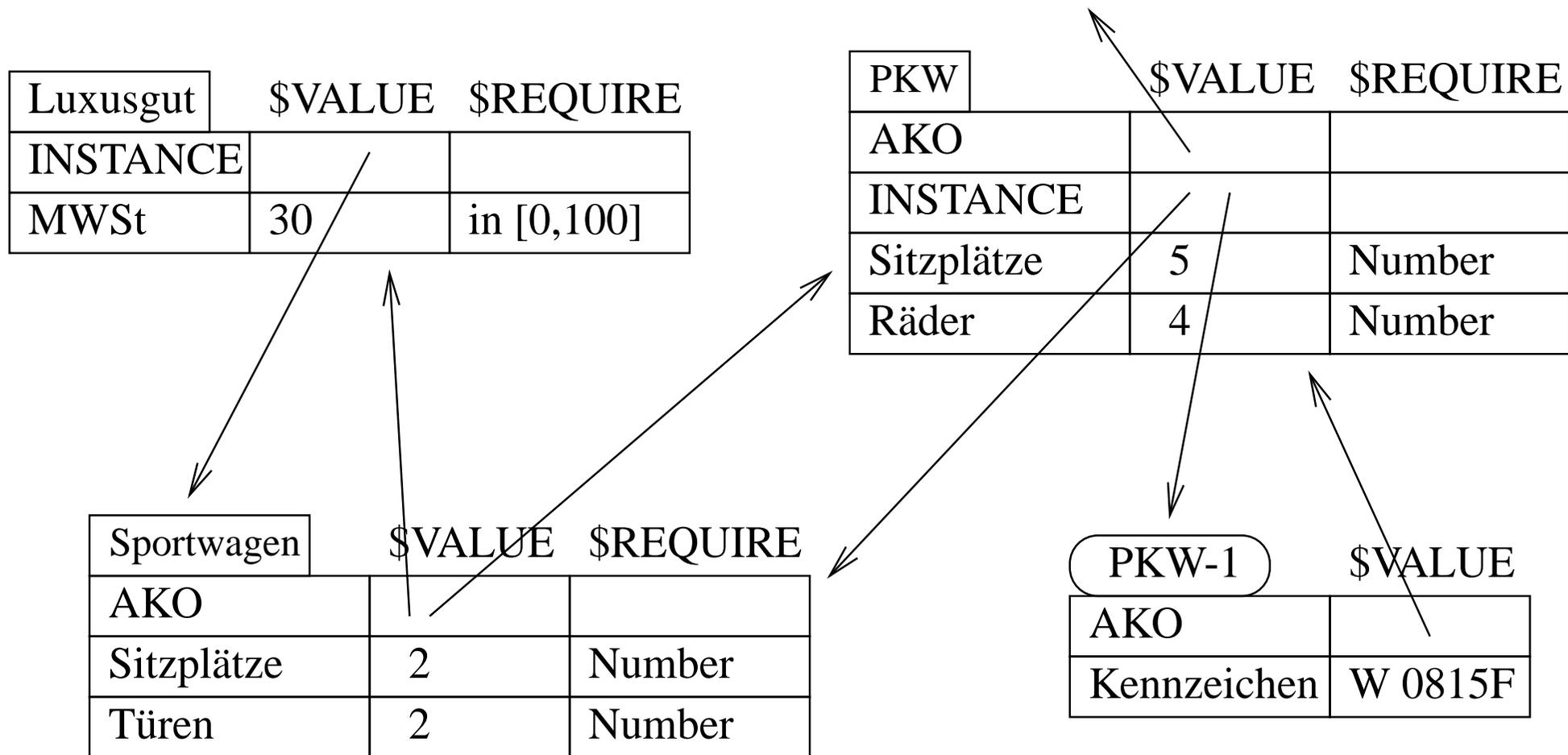


Abbildung 1: Einfaches FRL-Framesystem

Mögliche Slots:

\$VALUE: Enthält den eigentlichen Slotwert.

\$DEFAULT: Enthält Defaultwert.

\$REQUIRE: Liste von Bedingungen, die *\$value* erfüllen muß.

\$IF-ADDED, \$IF-REMOVED, \$IF-NEEDED: Code, der bei Eintragen, Löschen oder Verwenden von \$VALUE Werten ausgeführt werden muß (*Daemons*).

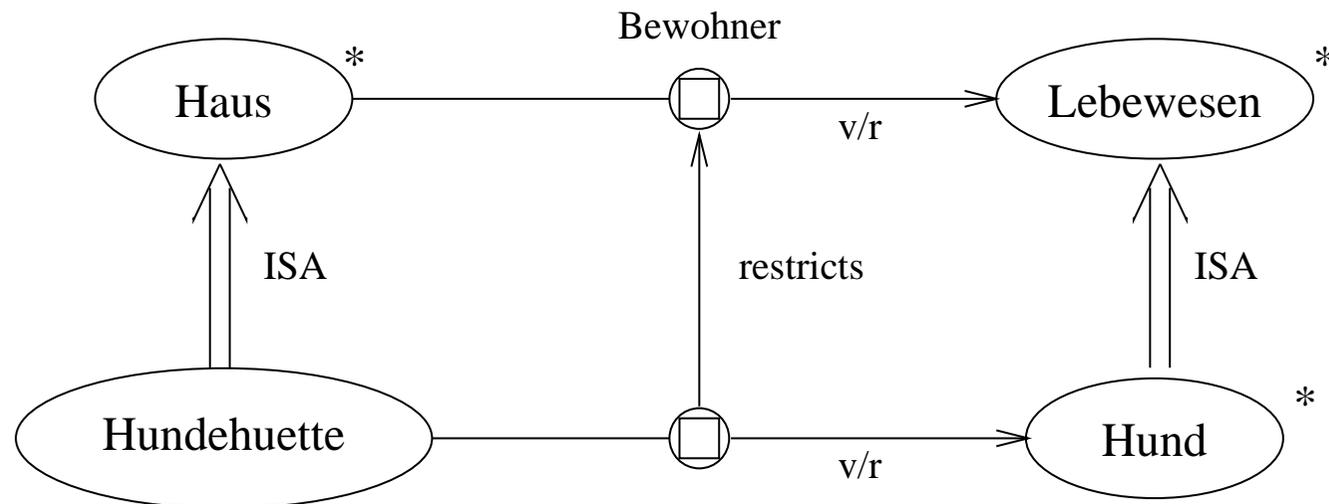
Ausführliche Beispiele mit Code werden in der LVA *Wissensbasierte Systeme* vorgestellt.

2.) Semantische Netze

Wissen wird in Form eines gerichteten Graphen repräsentiert.

Knoten: Konzepte; Kanten: Beziehungen (konzeptuelle Relationen).

Bekanntester Formalismus: KL-ONE (Brachman 1979)



Vorteile der Objektorientierten Wissensrepräsentation:

- “Realitätsnahe” Abbildung
- Deklarative Beschreibung struktureller Verwandtschaft von Objekten
- Leicht verarbeitbar (Graph-Algorithmen)

Nachteil: Eingeschränkte Möglichkeiten der *faktuellen* Schlußfolgerung.

Logikorientierte Wissensrepräsentation

Wissen wird in Form von *Fakten* und *Regeln* dargestellt.

vogel(taube)

vogel(kanarienvogel)

vogel(geier)

vogel(pinguin)

$(\text{vogel}(x) \wedge x \neq \text{pinguin}) \Rightarrow \text{kannfliegen}(x)$

Die folgenden neuen Fakten folgen logisch aus der Wissensbasis, und können abgeleitet werden:

kannfliegen(taube), kannfliegen(kanarienvogel), und kannfliegen(geier).

Die Regel

$$(\text{vogel}(x) \wedge x \neq \text{pinguin}) \Rightarrow \text{kannfliegen}(x)$$

entspricht der prädikatenlogischen Formel

$$\forall x [(\text{vogel}(x) \wedge x \neq \text{pinguin}) \Rightarrow \text{kannfliegen}(x)]$$

“ \models ” bezeichnet logische Folgerung, “ \neg ” Negation.

Wenn KB unsere Wissensbasis ist, gilt z.B.:

$$\text{KB} \models \text{kannfliegen}(\text{Geier})$$

$$\text{KB} \not\models \text{kannfliegen}(\text{Pinguin})$$

Achtung ! Das bedeutet nicht:

$$\text{KB} \models \neg \text{kannfliegen}(\text{Pinguin})$$

Es gilt vielmehr

$$\text{KB} \not\models \neg \text{kannfliegen}(\text{Pinguin})$$

Monotones Schließen

Die klassische Logik ist *monoton*. Das bedeutet, daß für alle Wissensbasen KB_1 und KB_2 gilt:

$$\text{Aus } KB_1 \models f \text{ folgt } KB_1 \cup KB_2 \models f$$

In Worten: Alles was aus KB_1 logisch folgt, folgt auch aus der erweiterten Wissensbasis $KB_1 \cup KB_2$.

In der Praxis schließt man nicht immer monoton!

Nichtmonotones Schließen

A: Auto wurde an Ort X abgestellt.

B: Auto steht an Ort X.

C: Auto wurde abgeschleppt.

*R*₁: Wenn man sein Auto an Platz X abgestellt hat, dann steht es (üblicherweise) dort.

*R*₂: Wenn das Auto abgeschleppt wurde, steht es nicht mehr am Platz. (Das ist äquivalent zu: $C \Rightarrow \neg B$.)

In der Praxis schließt man folgendermaßen ($|\approx$)

$\{A, R_1, R_2\} |\approx B$, aber $\{A, C, R_1, R_2\} \not|\approx B$.

Also nichtmonoton.

Wichtige Aufgaben der AI:

- Erforschung des menschlichen Schließens;
- Formalisieren des “Commonsense Reasoning”;
- Entwicklung von geeigneten nichtmonotonen Logiken;
- Studium, Vergleich und Anwendung solcher Logiken.

Ein bekanntes Beispiel ist die *Default Logik* (Reiter 1980).

Default Logic

A: Auto wurde an Ort X abgestellt.

B: Auto steht an Ort X.

C: Auto wurde abgeschleppt.

R_1 : Wenn man sein Auto an Platz X abgestellt hat, dann steht es (üblicherweise) dort.

R_2 : Wenn das Auto abgeschleppt wurde, steht es nicht mehr am Platz. ($C \Rightarrow \neg B$).

Regel R_1 wird ausgedrückt durch den Default d :

$$d = \frac{A : MB}{B}$$

d : Wenn A bereits gilt und wenn B zu schließen nicht zu Widersprüchen führt, dann schließe B .

Es gilt nun:

$\{A, d, R_2\} \mid \approx B$, aber $\{A, C, d, R_2\} \not\mid \approx B$.

Default Theorie für fliegende Vögel

vogel(taube).

vogel(kanarienvogel).

vogel(geier).

vogel(pinguin).

$\forall x: \text{pinguin}(x) \Rightarrow \neg \text{kannfliegen}(x).$

$$\frac{\text{vogel}(x) : M \text{kannfliegen}(x)}{\text{kannfliegen}(x)}$$

Inferenz

Beschäftigt sich mit der Frage, *wie* wir die gültigen Schlußfolgerungen herleiten können.

Prädikatenlogik unentscheidbar, daher oft Beschränkung auf entscheidbare Teilformalismen.

Effizientes Schließen möglich mit Fakten (z.B. $\text{vogel}(\text{Geier})$) und Implikationsregeln der Art:

$$(\text{vogel}(x) \wedge x \neq \text{pinguin}) \Rightarrow \text{kannfliegen}(x)$$

Zwei Hauptmechanismen des Schlußfolgerns: *forward chaining* und *backward chaining*.

Eine einfache Wissensbasis WB

mutter(wilma,anna)

mutter(anna,hans)

vater(hans,silvia)

vater(hans,martin)

vater(gregor,hans)

vater(paul, gregor)

vater(ernst,anna)

$\text{vater}(x,y) \rightarrow \text{elternteil}(x,y)$

$\text{mutter}(x,y) \rightarrow \text{elternteil}(x,y)$

$\text{elternteil}(x,y) \Rightarrow \text{vorfahre}(x,y)$

$\text{elternteil}(x,y) \wedge \text{vorfahre}(y,z) \Rightarrow \text{vorfahre}(x,z)$

Forward Chaining

Elementarer Ableitungsschritt:

- Finde Fakten, die zu den Prämissen einer Regel passen
- Ersetze die Variablen der Regel entsprechend
- Schließe das Konklusionsatom.

$$\text{vater}(x,y) \quad \Rightarrow \quad \text{elternteil}(x,y)$$

|

`vater(hans,silvia)`

Geschlossen wird: `elternteil(hans,silvia)`.

Forward Chaining (2)

Wiederhole einfache Ableitungsschritte, bis keine neuen Fakten mehr herleitbar sind.

Es ist herleitbar: `elternteil(wilma,anna)`, `elternteil(anna, hans)`,

`elternteil(hans, silvia)`

`elternteil(hans,martin)`, `elternteil(gregor, hans)`,

`elternteil(paul, gregor)`

`elternteil(ernst,anna)`,

`vorfahre(wilma,anna)`, `vorfahre(anna, hans)`,

`vorfahre(hans, silvia)`, `vorfahre(hans,martin)`,

`vorfahre(gregor, hans)`, `vorfahre(paul, gregor)`

`vorfahre(ernst,anna)`,

`vorfahre(wilma,hans)`, `vorfahre(anna,silvia)`, `vorfahre(wilma,silvia)`,...

Backward Chaining

Ich will überprüfen ob ein bestimmtes Faktum (Goal) aus WB herleitbar ist.

Z.B.: “vorfahre(wilma,hans)?”

Dann brauche ich nicht notwendigerweise *alle* herleitbaren Fakten inferieren.

Ich kann vielmehr vom Goal ausgehend Regeln rückwärts verfolgen und versuchen, eine Herleitung (Beweis) für das Goal zu konstruieren.

Backward Chaining Algorithmus

- Ist Goal Element von WB , fertig.
- Suche geeignete Regel, deren Kopf (Konklusion) mit dem Goal zusammenpaßt.
- Ersetze Variablen der Regel entsprechend.
- Beweise jedes Atom, das nun im Rumpf (Prämisse) der Regel auftritt (unter Berücksichtigung gleicher Variablen).

Goal: $\text{vorfahre}(\text{wilma}, \text{hans})$?

Versuche z.B. zuerst die Regel:

$\text{elternteil}(x, y) \Rightarrow \text{vorfahre}(x, y)$.

Subgoal: $\text{elternteil}(\text{wilma}, \text{hans})$. Unbeweisbar!

Versuche nun die Regel:

$\text{elternteil}(x, y) \wedge \text{vorfahre}(y, z) \Rightarrow \text{vorfahre}(x, z)$.

Ergibt subgoals: $\text{elternteil}(\text{wilma}, y)$, $\text{vorfahre}(y, \text{hans})$.

Diese sind ihrerseits beweisbar.

Forward Chaining: Verarbeitung der Regeln in Richtung des Implikationspfeiles.

Backward chaining: Verarbeitung der Regeln in der Gegenrichtung des Implikationspfeiles.

Meist ist backward chaining effizienter.

Backward chaining ist in einigen Fällen jedoch zeitaufwendiger, da das gleiche Subgoal öfters bewiesen werden muß (in Extremfällen sogar exponentiell oft).

Es existieren viele Verbesserungen der Forward Chaining Strategie.

System dl_v, am Institut für Informationssysteme entwickelt.