

# Wissenrepräsentation

In den Anfängen der Artificial Intelligence entsprach es der allgemeinen Ansicht, dass intelligentes Verhalten auf einer geschickten Suche nach Lösungen in einem bestimmten Problemraum beruht.

Es entstanden verschieden Paradigmen der AI:

**AI = generelle Symbolmanipulationsprinzipien auf denen Intelligenz fusst.**

„Ich benutze intelligente Werkzeuge (Methoden) , also denke ich“

Bald fand man aber heraus, dass eine solche Annahme nicht ganz richtig war, da der Mensch beim lösen seiner Probleme nicht nur auf allgemeine Methoden setzte, sondern auch mit Wissen arbeitet.

→ Wechsel zu Wissenbasierte System  
es gibt eine Wissensbasis, in der benötigtes Wissen gespeichert wird.

## **AI = Allgemeine Prinzipien + Wissensrepräsentation**

„Ich bin wissensbasiert, also denke ich“

Probleme:

Welches und wieviel?

Wie kann man benötigtes repräsentieren?

Wie kann man anwenden?

Genau diese Fragen sind von der Wissenrepräsentation zu klären.

### **Was ist Wissen?**

Über die Jahre haben sich einige Definitionen angesammelt.

- Wissen ist Abstraktion (Wiederhold 1986)
- Wissen ist eine Metapher. Wissen = Daten + Programm (Barr & Feigenbaum 1979)
- Wissen ist eine funktionale Grösse (Newell 1981)

### **Was ist Wissensrepräsentation ?**

Bestimmtes Wissen ist bestimmten Subeinheiten (z.B. Wissensbasis) eines Systems assoziiert, Wissen ist lokalisierbar im System.

## **Was sind nun die Aufgaben der Wissenrepräsentation?**

Entsprechend den obigen Informationen, Methoden zu entwickeln, mit deren Hilfe das Wissen in einem System dargestellt werden kann → Wissensbasis

2 Aufgabenbereiche:

- Formalismen
- Methoden zur Modellierung

### Repräsentationsformalismen

Jeder Formalismus besteht aus zwei Komponenten:

- Notation (Syntax)
- Inferenzkomponente (erlaubt die Verarbeitung)

Wesentlichste Verwendung ist das Schliessen ( Ziehen von Inferenzen) d.h. Ableiten neuer Fakten aus gespeicherten Daten.

### 3 Formen des Schliessens:

**Deduktion:** Explizitmachen implizit schon vorhandenen Wissens, Umformung bekannter Fakten in neu Fakten

Beispiel:

*Alle Personen die eine Afrikaurlaub machen, nehmen eine Flug.*

*Und*

*Herr Wagner macht einen Amerika Urlaub*

*Daraus folgt:*

*Herr Wagner nimmt einen Flug*

Dies entspricht den Modus Ponens der Logik (Aristoteles)

**Abduktion:** Finden einer möglichst realistischen Erklärung für bestimmte Fakten. Umkehrung des Modus Ponens.

Beispiel:

*Herr Wagner macht eine Bergtour*

*Und*

*Alle an den Bergen interessierten Personen machen eine Bergtour*

Mit Abduktion schliessen wir:

*Herr Wagner ist an den Bergen interessiert.*

Allerdings muss dieser Schluss im Gegensatz zur Deduktion nicht immer richtig sein.

Alternativen → nichtmonotnes Schliessen

**Induktion:** Generalisierung aus bekannten Fakten. Wichtig beim Maschinellen Lernen.

Beispiel:

*Vergleich Reiseveranstalter. ATA jeweils der günstigste → allen anderen Reisen bei ATA auch günstiger.*

## **Welche Eigenschaften ?**

***Expressivität:*** muss ausreichen sein um Wissen darzustellen, nicht zu viel sonst schlechte Verarbeitung

***Korrekt & Vollständig:*** Alle Schlüsse sind wahr und sind auch gezogen worden

***Effizienz:*** Wird durch Korrektheit und Vollständigkeit bestimmt.

Wichtig: Expressivität = Effizienz

## **Adäquatheitskriterien ?**

***Dichte der Darstellung.*** Ist eine nahezu redundanzfrei Darstellung möglich. Effizienz hängt wesentlich von der Grösse der Wissensbasis ab.

***Epistemologische Adäquatheit*** (Expressivität).

Wissensbasis muss es ermöglichen, alle für das System benötigten Fakten zu erschliessen.

***Ergonomische Aspekte.*** Ist die Wissensbasis erweiterbar, abänderbar. Zu grosse Datenmengen sind schlechter wartbar und es kann zu Fehlern kommen. Unterstützung durch geeignete Werkzeuge.

***Angabe von einer Semantik.*** Ist Möglichkeit gegeben, formal zu prüfen, ob alles implementiertes funktionstüchtig ist → Korrektheit & Vollständigkeit ?

***Heuristische Adäquatheit.*** Darstellung so gewählt, dass das Ziehen der Schlüsse möglichst einfach ist? Verteilung sollte so gewählt sein, dass häufig benötigte Fakten leichter erschlossen werden können.

## Welche Arten von Formalismen gibt es ?

### *Prozedurale Repräsentation:*

Wissensbasis = Programm

Für jeders Prädikat eine eigene Prozedur:

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

.  
.

```
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 das Wissen ändert.

## *Frames*

Idee der Frames hatte 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 frame. This is a remembered framework to be adapted to fit reality by changing details as necessary....

Wie sieht so was aus?

Siehe Abbildung 1

Frame: Menge von slots (attributen)

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 (Lisp) enthalten.

Vererbung von Attributen und Werten spielt eine wichtige Rolle

## **Propositionale Repräsentation**

Wissen in Form von Fakten. Form des Schliessen von Deduktion über Abduktion bis hin zum nicht monotonem Schliessen. Wichtig ist das Verhältnis zwischen

prozeduralem und deklarativen Wissen (= inhaltliche Eigenschaften spielen keine Rolle)

### *Semantische Netze*

Beruh auf der Idee der Assoziation. Zusammengehöriges Wissen, soll auch logisch nahe zueinander Repräsentiert werden.

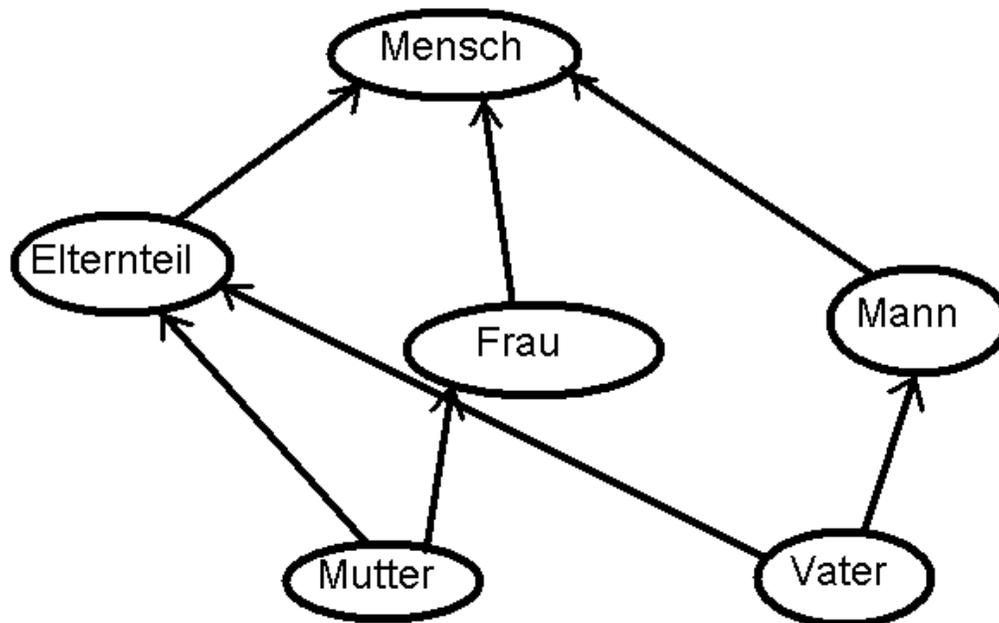
Darstellung als gerichtete Graphen

Knoten = Konzepten (Begriffen)

Kanten = Relationen

Wichtigste Relation : IS-A-Relation

Begriffe werden nach ihrer allgemeinheit geordnet



Wichtigste Form des Schliessens ist die Klassifikation, damit kann jedes neu Konzept in an den richtigen Platz gebracht werden.

Vorteile:            Realisätsnahe Abbildung  
                      Leichter verarbeitbar (Graphen)  
                      Deklarative Beschreibungen

Nachteil:            tw. Zu allgemein

### *Nicht monotone Vererbungsnetze*

Nicht monotones Schliesse (Default Schliessen) heisst, das der Erwerb neür Fakten alte Schlüsse ungültig machen kann.

Warum ist dies notwendig?

Unvollständiges Wissen: Wissen reicht nicht aus, somit wird ein Faktum angenommen, welches sich später als Falsch herausstellen kann.

z.B.

Willi ist ein Vogel. Normalerweise können Vögel fliegen  
→ Will kann fliegen.

Später erfährt man, dass Will aber ein Vogelstrauss ist.

## Änderung der Fakten

z.B.

Will ist ein Sittich. Damit gilt Will kann fliegen

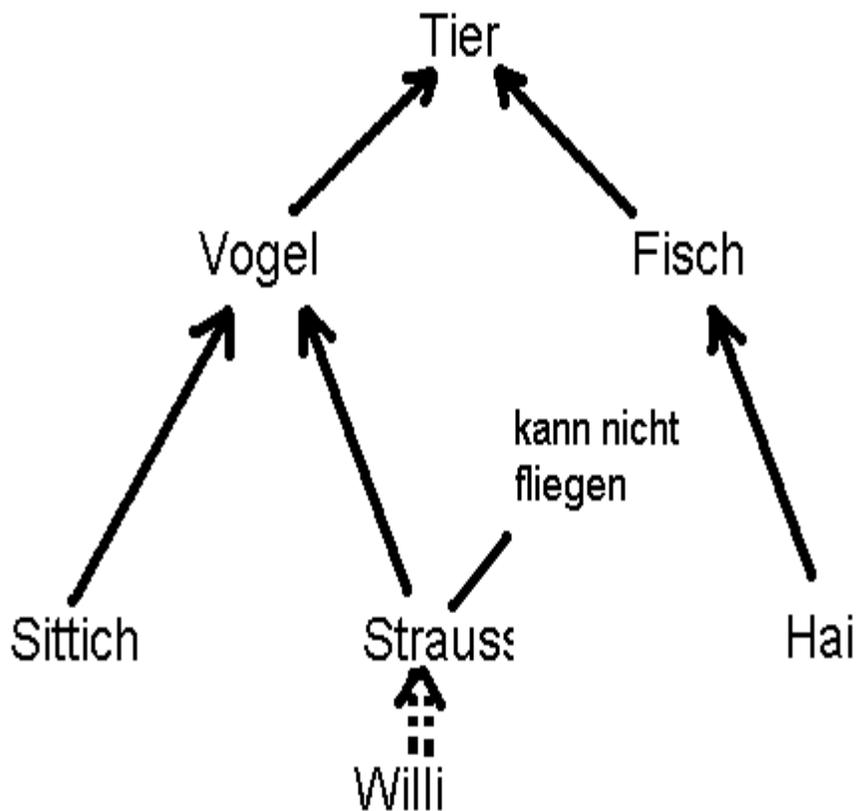
Durch eine Unfall bricht er sich den Flügel → kann nicht fliegen

Einfache Art des nicht monotonem Schliessens ist closed-world-assumption.

Verwendung in Datenbanken. System ist über alle positiven Fakten informiert. Vom Nichtvorhandensein eines Faktums kann also auf die Negation dessen geschlossen werden.

Beispiel:

Semantisches Netz, welches Ausnahmen zulässt. Fakten sind Defaults, die nur dann vererbt werden, wenn sie nicht überschrieben wurden.



## Logik

Ist das verbreitetste Beispiel für propositionale Repräsentation.

Verschieden Arten:

- Aussagenlogik
- Prädikatenlogik
- Modallogik
- Nichtmonotone Logik

Gemeinsame wichtige Eigenschaften:

- Vollständig deklarative Form, Operation rein auf struktureller Ebene.
- Es gibt immer eine formale Semantik (ist formal beschreibbar)

Jedoch:

Unvollständig und implizit.

Logik wird auf zwei verschiedenen Arten in der Wissensrepräsentation verwendet

- Formalismus
- Formale Beschreibung eines anderen Formalismus, um Aussagen über die Korrektheit und Vollständigkeit zu geben.

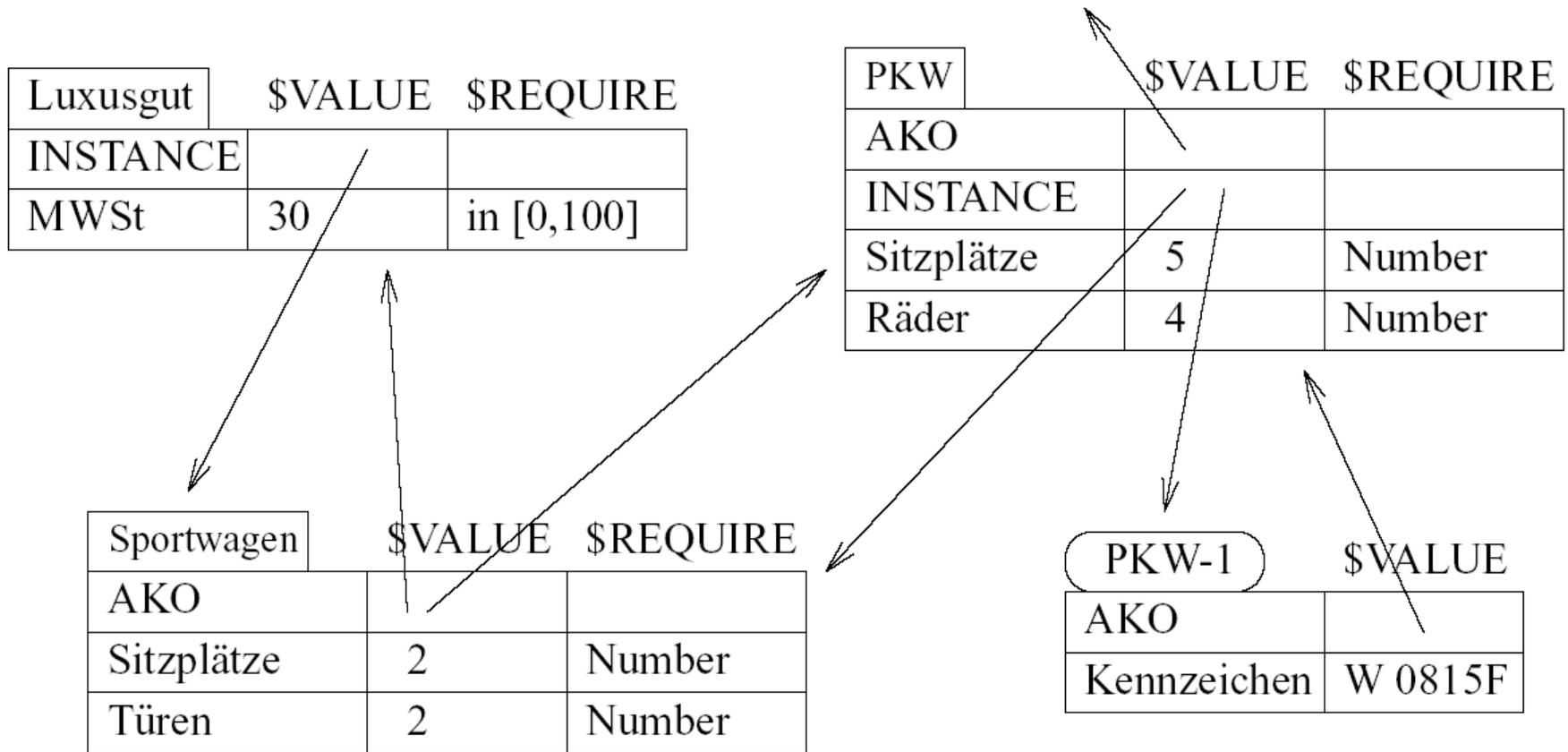


Abbildung 1: Einfaches FRL-Framesystem

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.