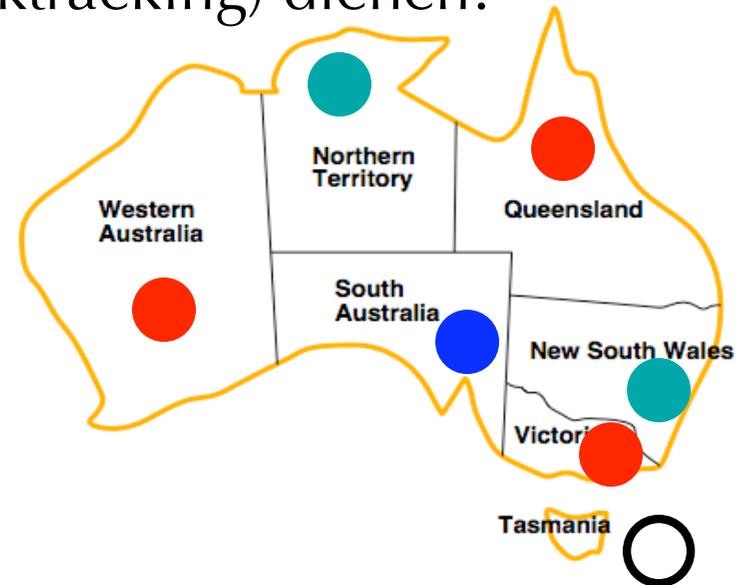


# Constraint Logic Programming (CLP, 1/2)

PROLOG kann als Constraintlöser (Backtracking) dienen!

## Beispiel

```
colorable(Wa,Nt,Sa,Q,Nsw,V,T) :-  
    diff(Wa,Nt), diff(Wa,Sa),  
    diff(Nt,Q), diff(Nt,Sa),  
    diff(Q,Nsw), diff(Q,Sa),  
    diff(Nsw,V), diff(Nsw,Sa),  
    diff(V,Sa).
```



```
diff(r,g).  
diff(r,b).  
diff(g,r).  
diff(g,b).  
diff(b,r).  
diff(b,g).
```

```
?- colorable(Wa,Nt,Sa,Q,Nsw,V,T).
```

```
Wa = r           Nt = g           Sa = b  
Q = r           Nsw = G          V = r  
T = _G184
```

Yes

# Constraint Logic Programming (CLP, 2/2)

PROLOG als Constraintlöser braucht:

- Diskrete Domains
- voll instanzierte Variablen

## Beispiel SWI-PROLOG

```
triangle(X,Y,Z) :-  
    X>=0, Y>=0, Z>=0,  
    X+Y>Z, Y+Z>X, X+Z>Y.
```

```
?- triangle(3,4,5).
```

Yes

```
?- triangle(3,4,U).
```

```
ERROR: Arguments are not  
sufficiently instantiated
```

## CLP-System würde antworten:

...

```
?- triangle(3,4,U).
```

```
7 > U > 1
```

Yes

In SWI-Prolog Zusatz CHR  
(Constraint Handling Rules)

# Fazit Prädikatenlogik und Resolution

- Prädikatenlogik 1. Stufe hat deutlich höhere Ausdrucksfähigkeit als Aussagenlogik. Damit ist sie als Basis für Formalismen der Wissensrepräsentation geeignet.
- Erfüllbarkeit in der PL ist unentscheidbar.
- Resolution funktioniert wie in der AL, muss aber zusätzlich Unifikation und Faktorisierung verwenden.
- Resolution (+Faktorisierung) in der PL ist korrekt & vollständig.
- Die Spezialisierungen von Resolution aus der AL können direkt weiter verwendet werden.
- Logikprogrammierung, CLP und Deduktionssysteme sind wichtige Anwendungen prädikatenlogischer Resolution.

# Ausblick in eine höhere Stufe

Da es Prädikatenlogik „1. Stufe“ gibt, gibt es natürlich auch Prädikatenlogik höherer Stufen.

## Beispiel

$$\forall R. [ \exists y. R(0, y) \wedge \forall x. (\exists y_1. R(x, y_1) \Rightarrow \exists y_2. R(x+1, y_2)) \\ \Rightarrow \forall x. \exists y. R(x, y) ]$$

Peano'sches Induktionsaxiom

# Gliederung

1. KI im Allgemeinen und in dieser Vorlesung
2. Heuristische Suche
3. Logik und Inferenz
- 4. Wissensrepräsentation**
5. Handlungsplanung
6. Lernen
7. Sprachverarbeitung
8. Umgebungswahrnehmung

- 1. Allgemeines & Überblick**
- 2. Kategorien & Objekte**
- 3. Zeit, Raum & Ereignisse**
- 4. Revidierbare Schlüsse**

## 1. Allgemeines & Überblick

# Wissen

Zusatzquelle: B. Nebel, Vorlesung Wissensrepräsentation, U. Freiburg

### Wissen

rheinland-pfälzische Stadt (Landkreis Altenkirchen), an der Sieg, 8800 Einwohner; Weißblechwerk, Kunststoffindustrie, Brauerei.

[www.wissen.de](http://www.wissen.de), 24.11.2004, Stichwortsuche „Wissen“, erster(!) Treffer

### Wissen

Gesamtheit aller organisierten Informationen mitsamt ihrer wechselseitigen Zusammenhänge, auf deren Grundlage ein (vernunftbegabtes) System handeln kann. Das Wissen erlaubt es einem solchen System – vor seinem Wissenshorizont und mit der Zielstellung der Selbsterhaltung – sinnvoll und bewusst auf Reize zu reagieren.

Wikipedia, 24.11.2004

# *The Knowledge Level (A. Newell, 1980)*

## **The Knowledge Level Hypothesis**

There exists a distinct computer system level, lying immediately above the symbol level, which is characterized by *knowledge* as the medium and [by] the *principle of rationality* as the law of behavior.

## **The Principle of Rationality**

If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action.

## **Knowledge**

Whatever can be ascribed to an agent such that its behavior can be computed according to the principle of rationality.

# Wissensrepräsentation

## Representation (Newell)

... the data structures that hold the problem and will be processed into a form that makes the solution available [and] the data structures that hold the world knowledge and will be processed to acquire parts of the solution or to obtain guidance in constructing it. ...

## Fragestellungen

- *Semantisch fundierte* Formalismen  
(z.B. Ausdruckstärke?, Inferenzverfahren?)
- Systeme (z.B. Effizienz? Korrektheit? Vollständigkeit?)
- Modellierung (z.B. Wartbarkeit?, Wiederverwendbarkeit?)

Hier weggelassen: **Wissenserhebung**  
(Knowledge Engineering, KADS-Methode, ...)

# Beispiele aus früheren Vorlesungsteilen

## Formalismen

- Aussagenlogik, Prädikatenlogik, Klausellogik
- Constraints

## Systeme

- SWI-PROLOG ( $\pm$  CHR)
- OTTER

## Modellierung

Bisher nicht thematisiert, sondern nur Mini-Vorlesungsmodelle!  
(In Russell/Norvig: Wumpus-Welt)

Die frühe KI sah Prädikatenlogik als die Lösung!  
Sie ist bis heute Vorbild für Semantik von Formalismen

# Forschungsinteressen/Schulen

- Interesse **Performanz**:  
Wie baue ich ein effizientes, wartbares Wissensbasiertes System?  
Zielrichtung: Informatik
- Interesse **Epistemologie**:  
Wo kommt Wissen her, und welche Arten davon gibt es?  
Zielrichtung: Philosophie, Logik
- Interesse **Kognition**:  
Wo kommt Wissen bei Menschen her und wie nutzen sie es?  
Zielrichtung: Kognitionswissenschaften

# Prozedural vs. deklarativ

- **Prozedurale** Semantik:

Die Bedeutung repräsentierten Wissens ergibt sich daraus, was ein Programm damit macht

↳ schwer zu kommunizieren und zu analysieren;  
potenziell effizienzsteigernd

z.B.: **begin real** x,y,z; **boolean** b; ... **end**

- **Deklarative** Semantik:

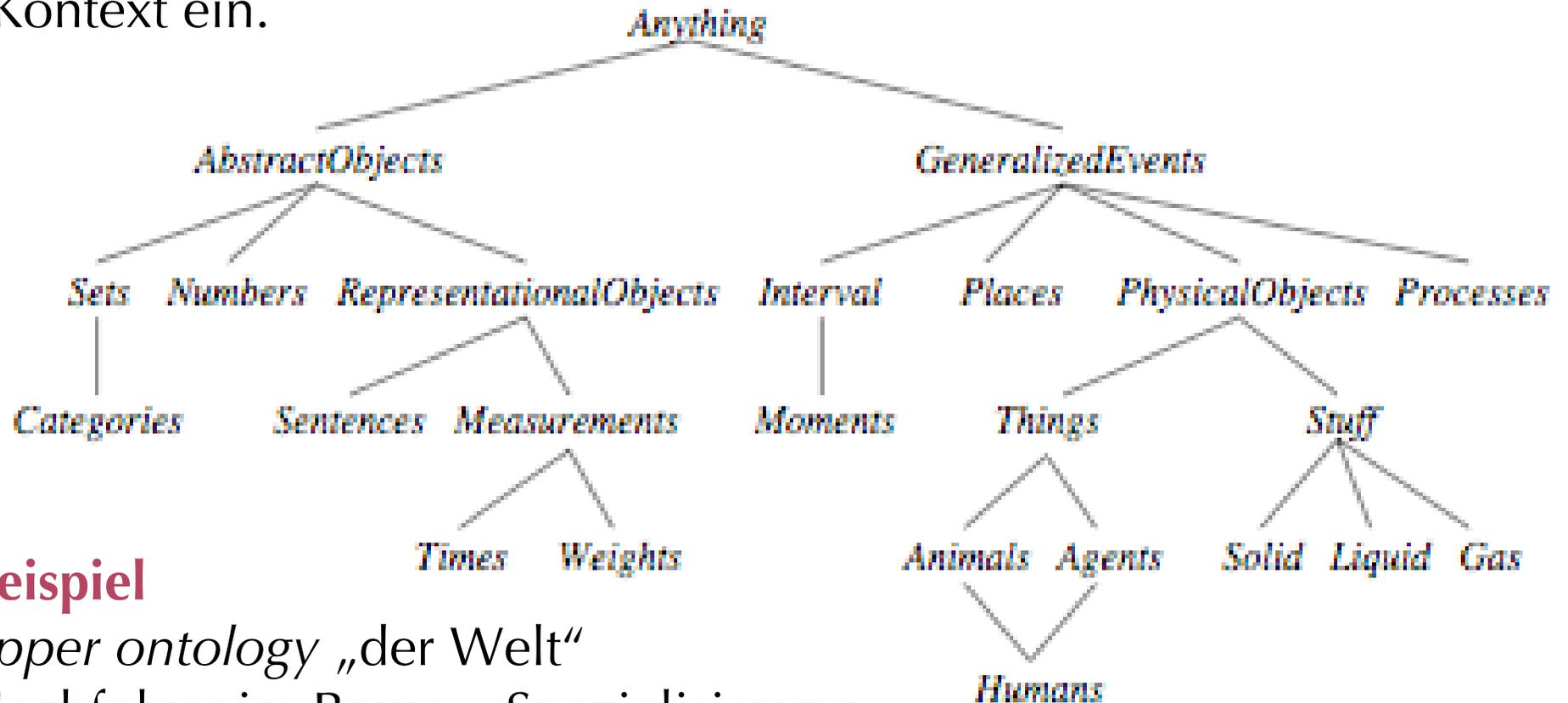
Abbildung der Sachverhalte auf abstrakte Sätze, die möglichst unabhängig von syntaktischen Varianten sind (Reihenfolge, ...)

↳ kommunizierbar, statisch verifizierbar und analysierbar.

z.B.: `triangle(x,y,z) :-`  
`x >= 0, y >= 0, z >= 0,`  
`x + y > z, y + z > x, x + z > y.`

# Ontologie

Problemübergreifende strukturierte (hierarchische) Darstellung eines Weltausschnitts; bettet Objekte, Aktionen, Ereignisse, ... in Kontext ein.



## Beispiel

*upper ontology* „der Welt“

Nachfolger im Baum » Spezialisierung

### Objektorientierte Programmierung (OOP)

Softwaredesign-Stil mit entsprechenden Sprachen:

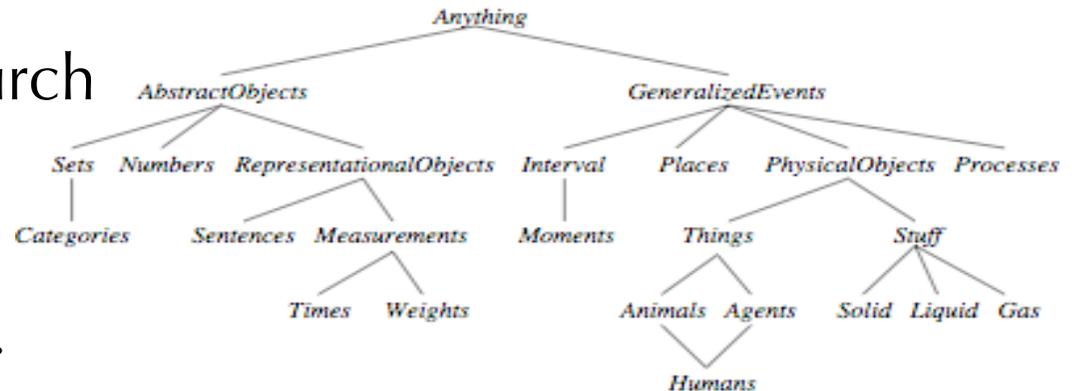
- „Objekte“ (Klassen, Instanzen) mit lokalem Zustand und lokalen Methoden und mit „Vererbung“ von Ober- zu Unterklassen
- Wurzeln (besonders 1970er Jahre):
  - Programmiersprachen Simula, SMALLTALK;
  - Wissensrepräsentationsformalismus Frames
- Heute u.a.: OO-Programmiersprachen (einschl. Java), OO Software-Engineering, UML, OO (Software) Design

# Grundbegriffe

- **Kategorien:** Klassen von Objekten, gemeinsame Eigenschaften  
z.B. *Basketball* (*rund*,  $\emptyset 9.5''$ , *orange*, ...)  
Logik-Analogon:  $\forall x. \text{Basketball}(x) \Rightarrow \text{Rund}(x)$   
Mengen-Sicht:  $\text{Basketball} \subseteq \text{Rund}$
- **Instanzen:** Individuelle Objekte, ggf. Vertreter einer Kategorie  
z.B.  $BB_9$  hat als *Basketball*-Instanz Eigenschaften *rund*,  $\emptyset 9.5''$ , ...  
Logik-Analogon:  $\text{Basketball}(BB_9)$   
Mengen-Sicht:  $BB_9 \in \text{Basketball}$
- **Vererbung:** Kategorien „erben“ von allgemeineren Kategorien  
Eigenschaften und möglicherweise Ausprägungen  
z.B. *Basketball* ist Teilkategorie von *Ball*. *Ball* hat  
Eigenschaft: *Form*, Ausprägung: *rund*; Eigenschaft  $\emptyset$  ohne Ausprägung  
Definiere *Basketball* als Unterkategorie von *Ball* mit  $\emptyset = 9.5''$ , Farbe *or*.  
Logik-Analogon:  $\forall x. \text{Basketball}(x) \Rightarrow \text{Ball}(x)$  (vgl.: **Sortierte** Logiken!)  
Mengen-Sicht:  $\text{Basketball} \subseteq \text{Ball}$

# Taxonomien

Objekthierarchie, die sich durch Zuweisung von Ober- und Unterkategorien sowie durch Angabe von Instanzen ergibt.



Aus Wissen über Objekteigenschaften schließe auf Zugehörigkeit zu Kategorien!

z.B.  $PQX_{17}$  ist *rund* mit  $\emptyset 9.5''$ , *orange* und *schmutzig*

→  $PQX_{17}$  ist ein *schmutziger Basketball*

Logisch:  $\forall x. Rund(x) \wedge Durchm(x)=9.5 \wedge Orange(x) [\wedge Ball(x)]$

$\Rightarrow Basketball(x)$

# Weitere Relationen zwischen Kategorien

Zwischen Kategorien (und damit zwischen ihren Unter-Kategorien und ihren Instanzen) können weitere vordefinierte Relationen bestehen:

- Disjunktheit  
z.B.: *Disjoint*({*Basketballs*, *Soccerballs*})
- Vollständige Überdeckung (möglicherweise mit Schnitt)  
z.B.: *ExhaustiveDecomposition*({*Americans*, *Canadians*, *Mexicans*},  
*NorthAmericans*)
- Disjunkte Überdeckung  
z.B.: *Partition*({*Males*, *Females*}, *Humans*)

# Beschreibungslogiken

*description logics, terminological logics*

- Komplexe Beschreibungen von Kategorien und Instanzen
- Abstrakte Terminologie und konkrete Objekte separierbar (TBox, ABox)
- Effiziente Schlüsse über Kategorien und Objekte
- „syntaktische Variante“ effizient entscheidbarer Untermengen von Prädikatenlogik

# Konzepte und Rollen

- **Konzepte:** Entsprechen Kategorien  
z.B. *Basketball(.)*  
logische Semantik: Einstellige Prädikate, z.B. *Basketball(x)*
- **Rollen:** Entsprechen Relationen zwischen Objekten  
z.B. *siblings(.,.)*  
logische Semantik: Zweistellige Prädikate, z.B. *siblings(x,y)*
- **Boolesche Operatoren auf Konzepten:**  $C \otimes D$ ,  $C \oplus D$ ,  $\sim C$   
z.B. *Human* $\otimes$ *Female*, *Father* $\oplus$ *Mother*,  $\sim$ *Male*  
logische Semantik:  $C(x) \wedge D(x)$ ,  $C(x) \vee D(x)$ ,  $\neg C(x)$
- **Boolesche Operatoren auf Rollen:**  $r \otimes s$ ,  $r \oplus s$ ,  $\sim r$   
z.B. *siblings* $\otimes$ *equal\_birthday*, etc.

# Rollenrestriktionen

- **Einschränkungen der „Füller“ von Rollen:** Allquantifizierung  
z.B. *Mother-without-Daughter*  
Grundidee: Verwende Quantoren:  $Mother \otimes \forall has-child.Man$   
logische Semantik:  $\forall y. r(x,y) \Rightarrow C(y)$   
z.B.  $Mother(x) \wedge \forall y.[has-child(x,y) \Rightarrow Man(y)]$
- **Spezialisierung der „Füller“ von Rollen:**  
Existenzquantifizierung  
z.B. *Grandmother* (Mutter, die ein Kind hat, das Elternteil ist)  
Grundidee: Verwende Quantoren:  $Mother \otimes \exists has-child.Parent$   
logische Semantik:  $\exists y. r(x,y) \wedge C(y)$   
z.B.  $Mother(x) \wedge \exists y.[has-child(x,y) \wedge Parent(y)]$

# Mehr zu Rollen

- **Numerische Einschränkungen der „Füller“ von Rollen:**  
z.B. Mutter mit 3 oder mehr Kindern,  $Mother_{\otimes}(\geq 3 \text{ has-child})$   
logische Semantik: offensichtlich (handwerklich aufwändig)
- **Inverse Rollen:  $r^{-1}$**   
z.B. „Kinder reicher Eltern“:  $\exists \text{has-child}^{-1}.Rich$   
logische Semantik:  $r^{-1}(x,y) \equiv r(y,x)$
- **Komposition:  $r \circ s$**   
z.B. Hat Enkelkind,  $\text{has-child} \circ \text{has-child}$   
logische Semantik:  $r(x,z) \equiv \exists y.[r(x,y) \wedge r(y,z)]$

# TBox

- Begriffswissen (*terminological knowledge*):  
Kategorienhierarchie, Eigenschaften von Kategorien  
z.B.: Basketball ist Teilkategorie von Ball. Ball hat Eigenschaft ...
- Zwei Arten **Terminologischer Axiome**:  
 $A \stackrel{!}{=} C$ : logische Semantik:  $\forall x. A(x) \Leftrightarrow C(x)$   
 $A \sqsubseteq C$ : logische Semantik:  $\forall x. A(x) \Rightarrow C(x)$
- Keine Mehrfachdefinitionen von A;  
keine zyklischen Definitionen  
(A wird definiert durch B wird definiert ... durch A)
- **Modell** einer TBox: Nichtleere Interpretation, die alle TBox-Axiome erfüllt.

# Beispiel-TBox

Male  $\doteq \sim$  Female  
Human  $\sqsubseteq$  Living\_entity  
Woman  $\doteq$  Human  $\otimes$  Female  
Man  $\doteq$  Human  $\otimes$  Male  
Mother  $\doteq$  Woman  $\otimes$   $\exists$ has-child.Human  
Father  $\doteq$  Man  $\otimes$   $\exists$ has-child.Human  
Parent  $\doteq$  Father  $\oplus$  Mother  
Grandmother  $\doteq$  Woman  $\otimes$   $\exists$ has-child.Parent  
Mother-without-daughter  $\doteq$  Mother  $\otimes$   $\forall$ has-child.Male  
Mother-with-many-children  $\doteq$  Mother  $\otimes$  ( $\geq 3$ has-child)