Universität Bielefeld

1 Symbolische Repräsentation

2. Vorlesung: Repräsentation mit Logik, semantischen Netzen und Frames

Methoden der Künstlichen Intelligenz

Ipke Wachsmuth WS 2008/2009

Interne Repräsentation

Zentral für „reasoning“: Interne symbolische Repräsentation und Symbolverarbeitung.

- Repräsentation ganz allgemein:

  Eine idealisierte Darstellung der Welt (nicht notwendig symbolisch)

- Interne, symbolische Repräsentation:

  erfordert eine einheitliche (Symbol-)Sprache, in der ein Agent Aussagen über die Welt ausdrücken und manipulieren kann.

- Gut geeignet

  für symbolische Repräsentationen sind Logik-Sprachen, jedoch sind Vorbereitungen zu treffen...

Referenten explizit machen

Natürliche Sprache ist oft mehrdeutig:

Der Hund saß auf dem Tisch.
Er bellte.

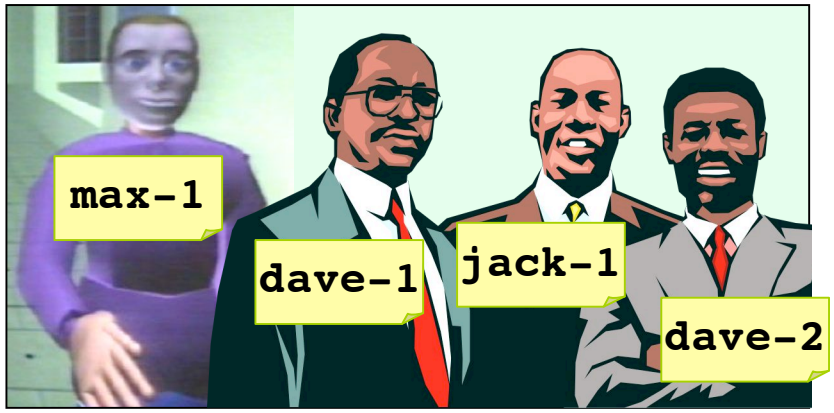
Der Hund (r1) saß auf dem Tisch (r2).
Er (r1) bellte.

(Nun ist klar, wer hier gebellt hat.)

Es kann den gleichen Namen mehrfach geben:

Das soll Dave machen!

Welcher „Dave“?



Referentielle Eindeutigkeit

☆ **1. Forderung: Symbolische Repräsentationen müssen Bezüge auf Referenten explizit machen!**

D.h. alle Mehrdeutigkeiten im Bezug auf Referenten müssen in der internen Repräsentation eliminiert werden:

- **jedes Individuum bekommt einen eindeutigen Namen**
- **d.h. nur ein Individuum pro Name vorhanden**

Also: statt mehrerer "Daves": `dave-1, dave-2` usw.

Solche eindeutigen Namen heißen *Instanzen* oder auch *Token*.

Semantische Eindeutigkeit

☆ **2. Forderung: Alle Symbole einer internen Repräsentation müssen eindeutig ("unambig") sein!**

Unterschiedliche Symbole implizieren unterschiedliche Semantik.

Beispiele für semantische ("Wortsinn"-) Mehrdeutigkeit:

Hans bringt das Geld auf die Bank. `[Geldbank]`

Hans setzt sich auf die Bank. `[Sitzbank]`

Jack caught a ball. `[catch-object]`

Jack caught a cold. `[catch-illness]`

(Wer eine Erkältung eingefangen hat, muss z.B. niesen.)

Funktionale Eindeutigkeit

☆ **3. Forderung: Interne Repräsentationen müssen die funktionalen Rollen eindeutig ausdrücken!**

Petra fängt die Keule.
Die Keule fängt Petra.
Die Keule wird von Petra gefangen.

Wer fungiert als Fänger? **Wer oder was als Gefangenes?**

Fazit: Symbolische Repräsentationen müssen also in mehrfacher Hinsicht eindeutig sein:

- **referentiell**
- **semantisch**
- **funktional**

Vom Satz zur Repräsentation

einfaches Beispiel

Jack caught a ball.

Referenten disambiguiert

jack-2 caught **ball-5.**

Wortsinn disambiguiert

jack-2 catch-object ball-5 .

Klammern als Begrenzer

(jack-2 catch-object ball-5)

Prädikat als Präfix

(catch-object jack-2 ball-5)

Prädikate, Formeln, Assertionen

Für das sprachliche catch wurde ein (2-stelliges) Prädikat `catch-object` in der Repräsentation eingeführt:

```
(catch-object jack-2 ball-5)
```

P(a,b)

Prädikat

Argumente

Mit einer *Formel* wird ein Faktum über eine oder mehrere Entitäten (Einzeldinge) ausgedrückt, in diesem Fall eine Fangen-Beziehung zwischen einem gewissen Jack und einem gewissen Ball.

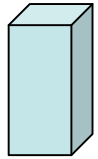
- **Assertionen sind Formeln, die man als <u>gegeben</u> ansieht (als Element einer aktuellen internen Repräsentation)**

Satz und Repräsentation

Im allgemeinen wird ein Satz durch <u>mehrere</u> Formeln repräsentiert:

Jack caught a blue block.

```
(catch-object jack-1 block-1)
(inst block-1 block)
(color block-1 blue)
```

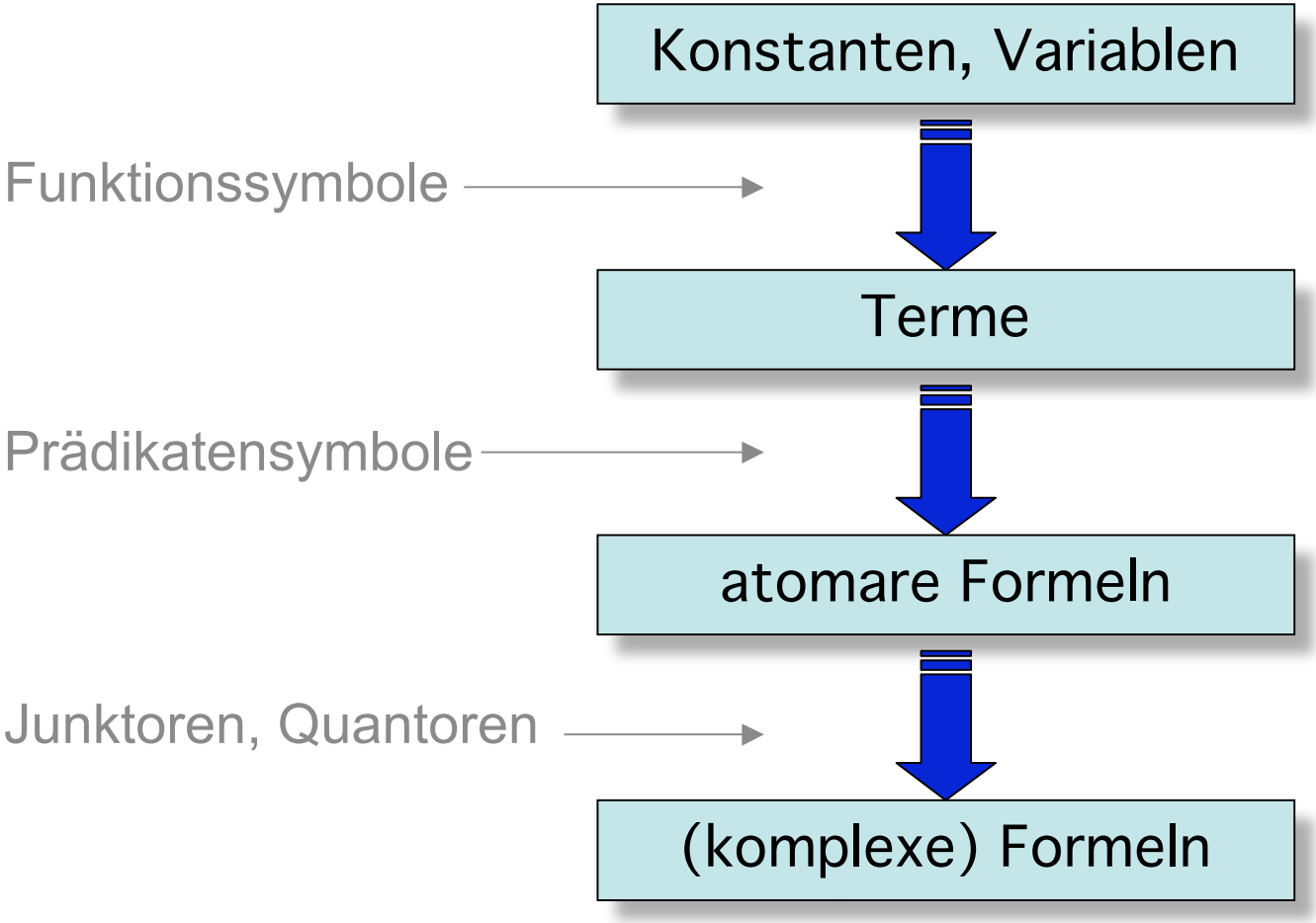


Prozesse auf internen Repräsentationen dienen dazu, aus bekannten Fakten neue zu gewinnen: ***Inferenzbildung***

Häufigst gebrauchter Inferenzbegriff: ***Deduktion*** –> TEIL 3

Solche Prozesse lassen sich in der Prädikatenlogik modellieren.
(In der Regel werden wir Prädikatenlogik 1. Stufe betrachten.)

Syntax der Prädikatenlogik (1. Stufe)



Es ist auch möglich, Konstanten als null-stellige Funktionen einzuführen (vgl. Skript Theoretische Informatik)

Prädikate und Argumente

In der KI ist die Prädikatenlogik ein seit langem erprobtes Mittel, Wissensinhalte auszudrücken.

Es dürfen beliebige Prädikate eingeführt werden.

- **Jedes Prädikat benötigt ein oder mehrere Argumente.**
- **Prädikatargumente werden durch Terme gefüllt.**

Terme können sein:

- **Konstanten** `block-1, jack-1, blue`
- **Variablen** `x, y, z`
- **Funktionsanwendungen** `(son-of jack-1)`

Freiheit in der Modellierung:

"block-1
ist ein Block"

Ein Prädikat mit gefüllten Argumenten (in der richtigen Zahl) ist der einfachste Typ einer Formel: <u>atomare Formel.</u>

z.B. `(inst block-1 block)` — Prädikat: `inst`; Argumente: `block-1`, `block`

`(block block-1)` — Prädikat: `block`; Argument: `block-1`

Junktoren und Quantoren

werden gebraucht, um aus gegebenen Formeln weitere (komplexe) Formeln zu konstruieren:

`and`	$\cong \&, \wedge$
`or`	$\cong \vee$
`not`	$\cong \neg$
`if`	$\cong \rightarrow$
`forall`	$\cong \forall$
`exists`	$\cong \exists$

Wenn p und q Formeln sind, sind auch

(and p q)
(or p q)
(not p)
(if p q) **Formeln.**

Ist x eine Variable und p eine Formel, dann sind auch

(forall(x) p)
(exists(x) p) **Formeln.**

p , q stehen hier als Platzhalter für irgendwelche Formeln.
(ASCII-Schreibweise bereitet maschinelle Verarbeitung vor)

Junktoren und Quantoren (Beispiele)

p q

```
(and (inst block-1 block) (color block-1 yellow))
```

"Block-1 ist ein gelber Block/Klotz."

```
(if (supported-by block-2 block-1)(on block-2 block-1))
```

"Wenn Block-2 von Block-1 getragen wird, dann steht Block-2 auf Block-1."

```
(forall(z) (if (inst z elephant)(color z gray)))
```

"Alle Elefanten sind grau."

```
(forall(x) (if (inst x person)
               (exists(y)(and (inst y head)(partof y x)))))
```

"Jede Person hat einen Kopf."

Slot-Assertion-Notation

Zweck: Ausdruck funktionaler Beziehungen

Auch dies sind prädikatenlogische Repräsentationen, die jedoch (durch die Slot-Prädikate) mehr ausdrücken als die oberen: *Funktionale Struktur*

Beispiele.

```
(catch-object jack-2 ball-5)
(catch-object petra-1 keule-3)
```

Prädikat

Argumente (slots)

werden repräsentiert als: **(immer noch Prädikatenlogik)**

```
(inst catch-22 catch-object)
(catcher catch-22 jack-2)
(caught catch-22 ball-5)

(inst catch-23 catch-object)
(catcher catch-23 petra-1)
(caught catch-23 keule-3)
```

Slot-Prädikate

Slot-and-Filler-Notation (->Frames)

Aus einer Menge von Fakten (Assertionen) wird ein "objektzentriertes" Format.

Objekt hier:

Das "catch-object-Ereignis `catch-22`"

Die verschiedenen Slot-Assertions werden zu einem strukturierten Ausdruck kombiniert:

```
Aus     (inst catch-22 catch-object)
        (catcher catch-22 jack-2)
        (caught catch-22 ball-5)

wird    (catch-object   catch-22
                        (catcher jack-2)
                        (caught  ball-5))
```

Allgemeine Struktur:

```
(catch-object   <token>
                (catcher <token>)
                (caught  <token>))
```

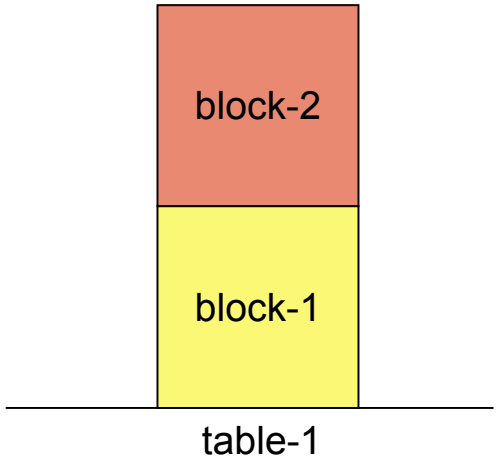
Slot Filler

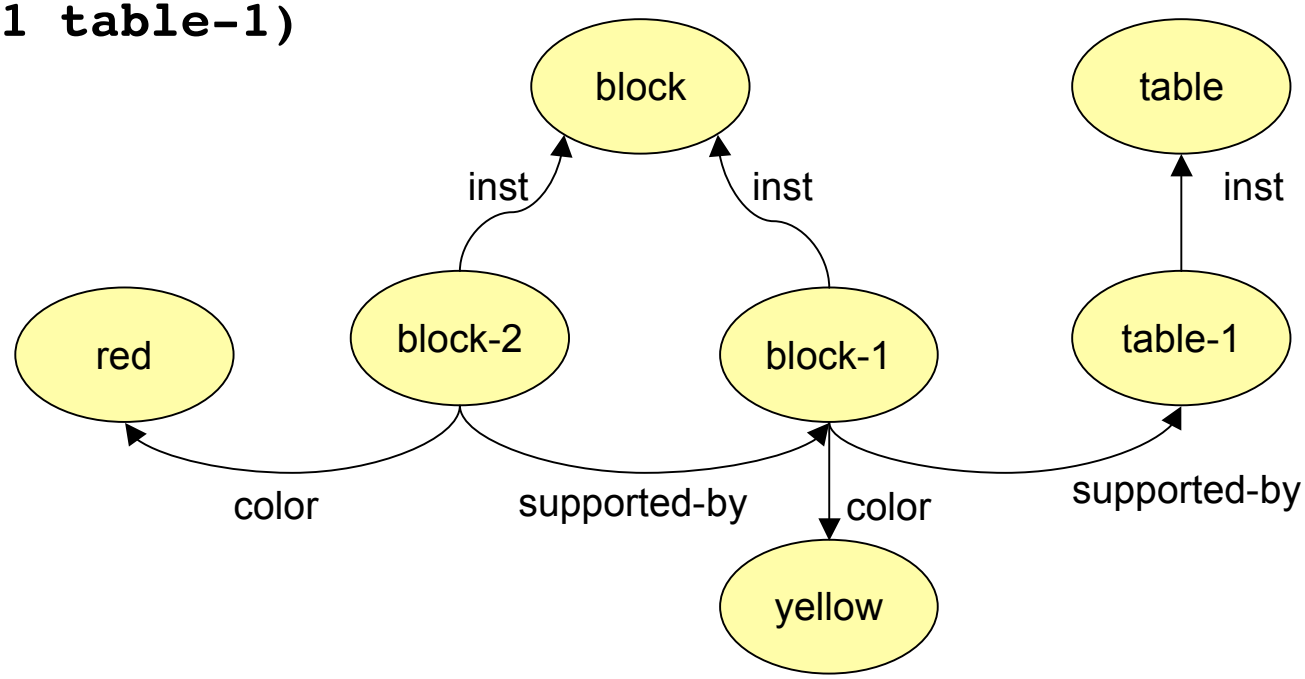
(objektzentriertes Format!)

Repräsentation einer Szene, 2 Arten

```
(inst block-2 block)
(color block-2 red)
(supported-by block-2 block-1)
(inst block-1 block)
(color block-1 yellow)
(supported-by block-1 table-1)
(inst table-1 table)
```

- als Menge logischer Formeln
- als semantisches Netz





Alternative Notationen

Semantische Netze (auch: "assoziative Netze") und prädikatenlogische Formeln repräsentieren gleiche Information in verschiedenem Format:

Knoten entsprechen Termen
markierte gerichtete Kanten entsprechen Prädikaten

d.h. es sind *alternative Notationen* für den gleichen Inhalt, nicht prinzipiell verschiedene Repräsentationen!

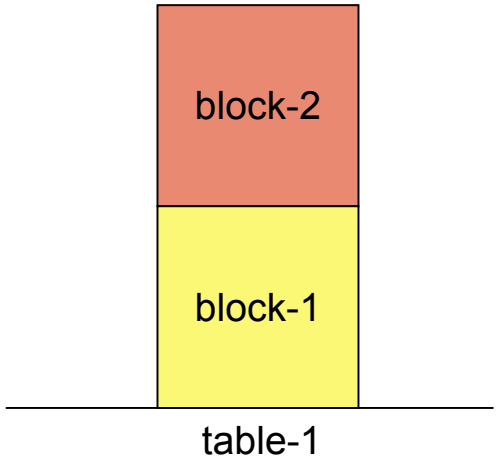
☆ **WAS IST ABER ANDERS?**
Zusätzlich enthalten semantische Netze *pointer* (und manchmal auch *back pointer*), die den Zugriff auf assoziierte Information (z.B. Instanzen) einfach machen:

INDEXING

[ähnlich: Frames]

Repräsentation einer Szene, 3. Art

```
(inst block-2 block)
(color block-2 red)
(supported-by block-2 block-1)
(inst block-1 block)
(color block-1 yellow)
(supported-by block-1 table-1)
(inst table-1 table)
```



"alternative Notationen"

- als Frames (slot-and-filler-Notation)

Frame	Attribute (slots)	Werte (fillers)
block-2 :	inst :	block
	color :	red
	supported-by :	block-1
	...	...

Frame	Attribute (slots)	Werte (fillers)
block-1 :	inst :	block
	color :	yellow
	supported-by :	table-1
	...	...

Frame	Attribute (slots)	Werte (fillers)
table-1 :	inst :	table
	color :	
	supported-by :	
	...	...

ISA-Hierarchie und Vererbung

- **Schlüsselkonzept der KI aus der Tradition semantischer Netze**
- **Instanzen "erben" Eigenschaften, die Individuenmengen (Klassen) zugeschrieben werden.**
- **Dies kann sich über ganze ISA-Hierarchien fortpflanzen.**

ISA:
“is a”
“ist ein”

INHERITANCE OF PROPERTIES

Zweck: ***Ökonomie in der Wissensrepräsentation***

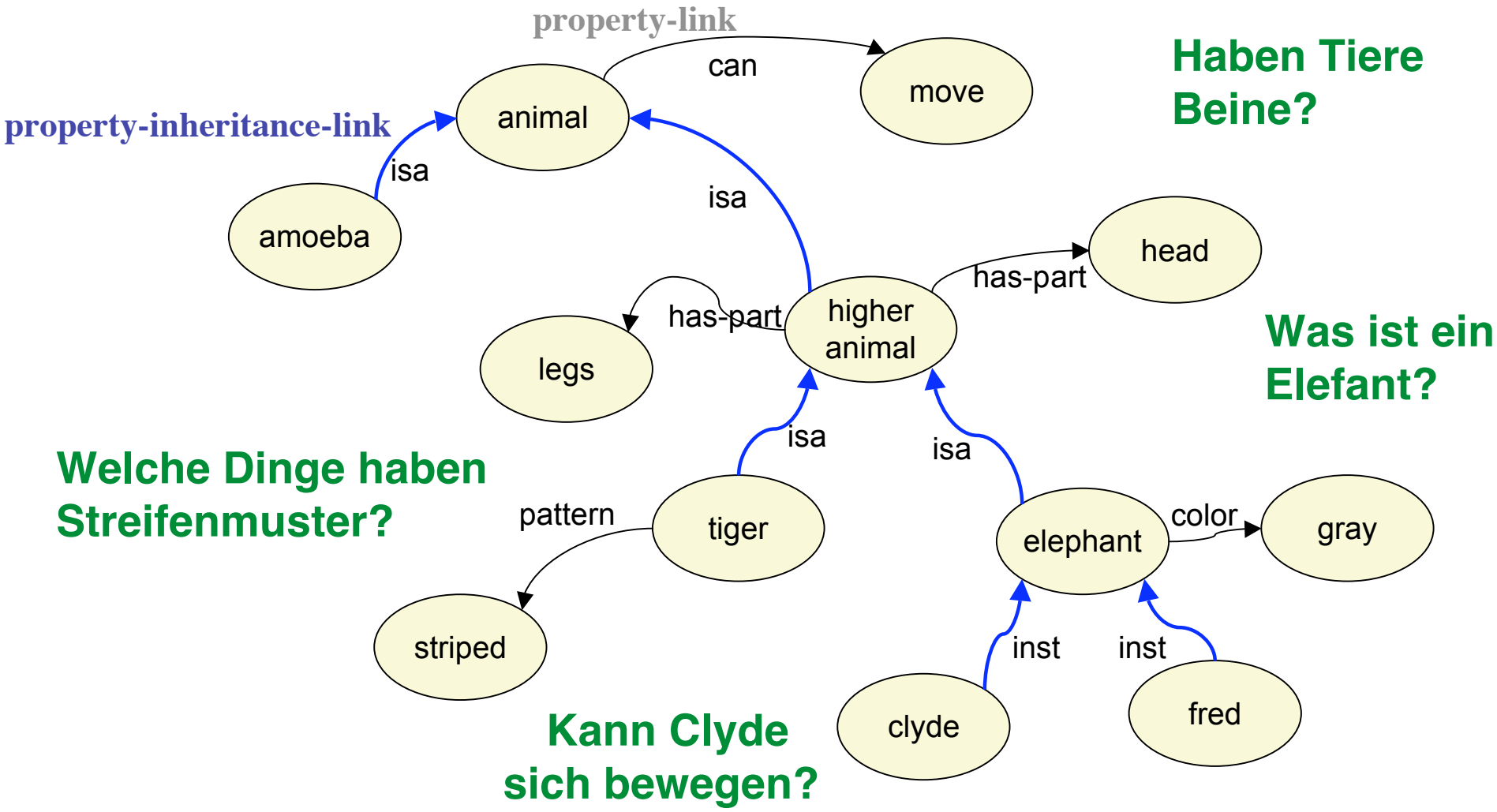
–> Suche entlang von inst- und isa-links, um nicht direkt verfügbare Information zu gewinnen (durch Vererbung).

inst:
“Instanz von”

inst **entspricht** $\in$

isa **entspricht** $\subseteq$

Beispiel einer ISA-Hierarchie



Haben Tiere Beine?

Was ist ein Elefant?

Welche Dinge haben Streifenmuster?

Kann Clyde sich bewegen?

Allgemeine vs. individuelle Eigenschaften

ACHTUNG – unterscheide:

Property-links von Klassen-Knoten eines semantischen Netzes `(dog,mammal)`:

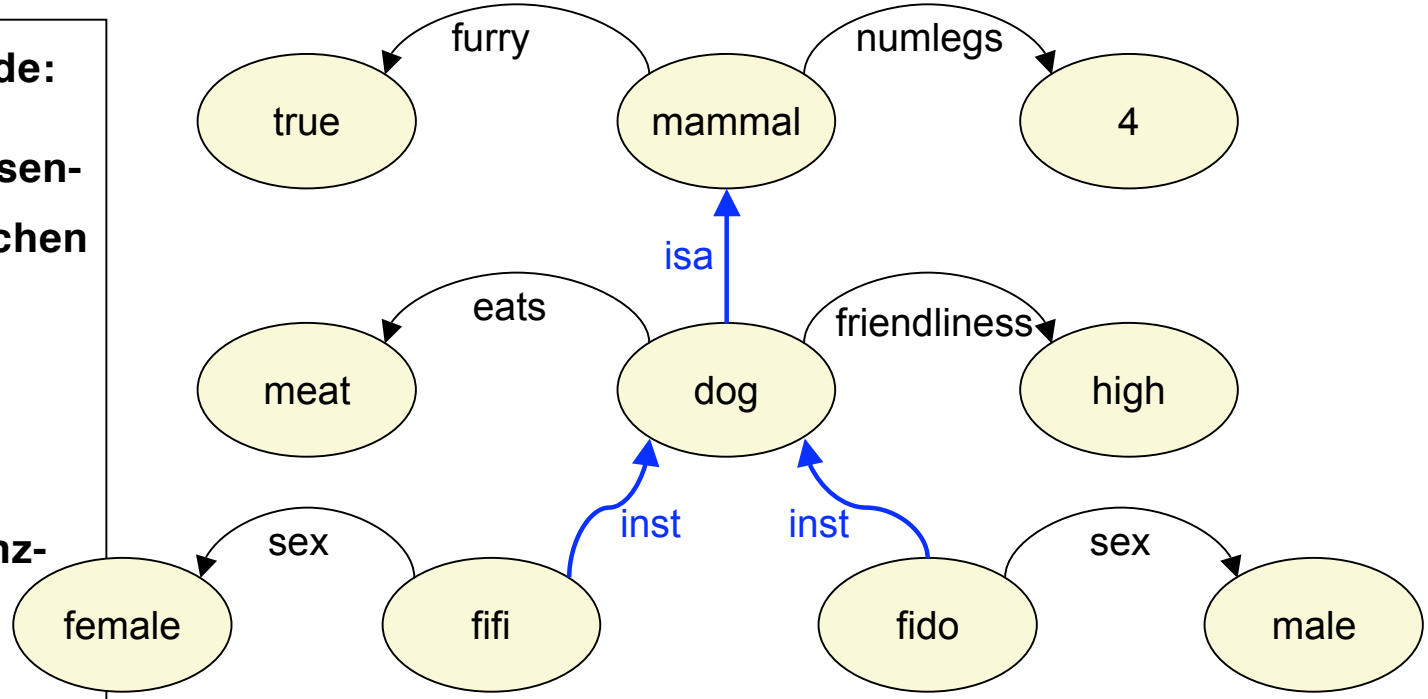
implizit allquantifizierte Assertionen *

Property-links von Instanz-Knoten `(fido,fifi)`:

assertierte Fakten für Individuen

z.B. `(sex fifi female)`

Type *versus* Token!

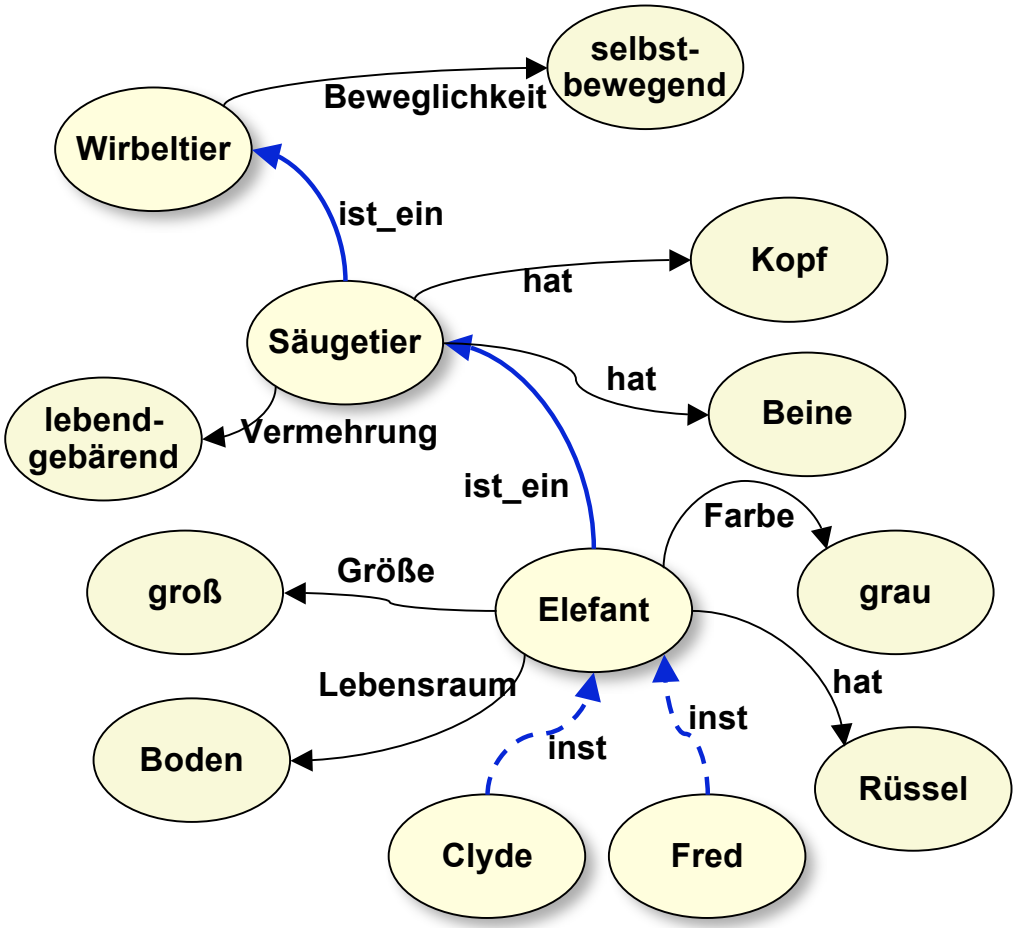


*Beispiel: prädikatenlogische Rekonstruktion der dog-properties

```
(forall(x)(if (inst x dog)
              (and (friendliness x high)
                   (eats x meat))))
```

Vererbung in sem. Netzen und Frames

(etwas anders modelliert als oben)

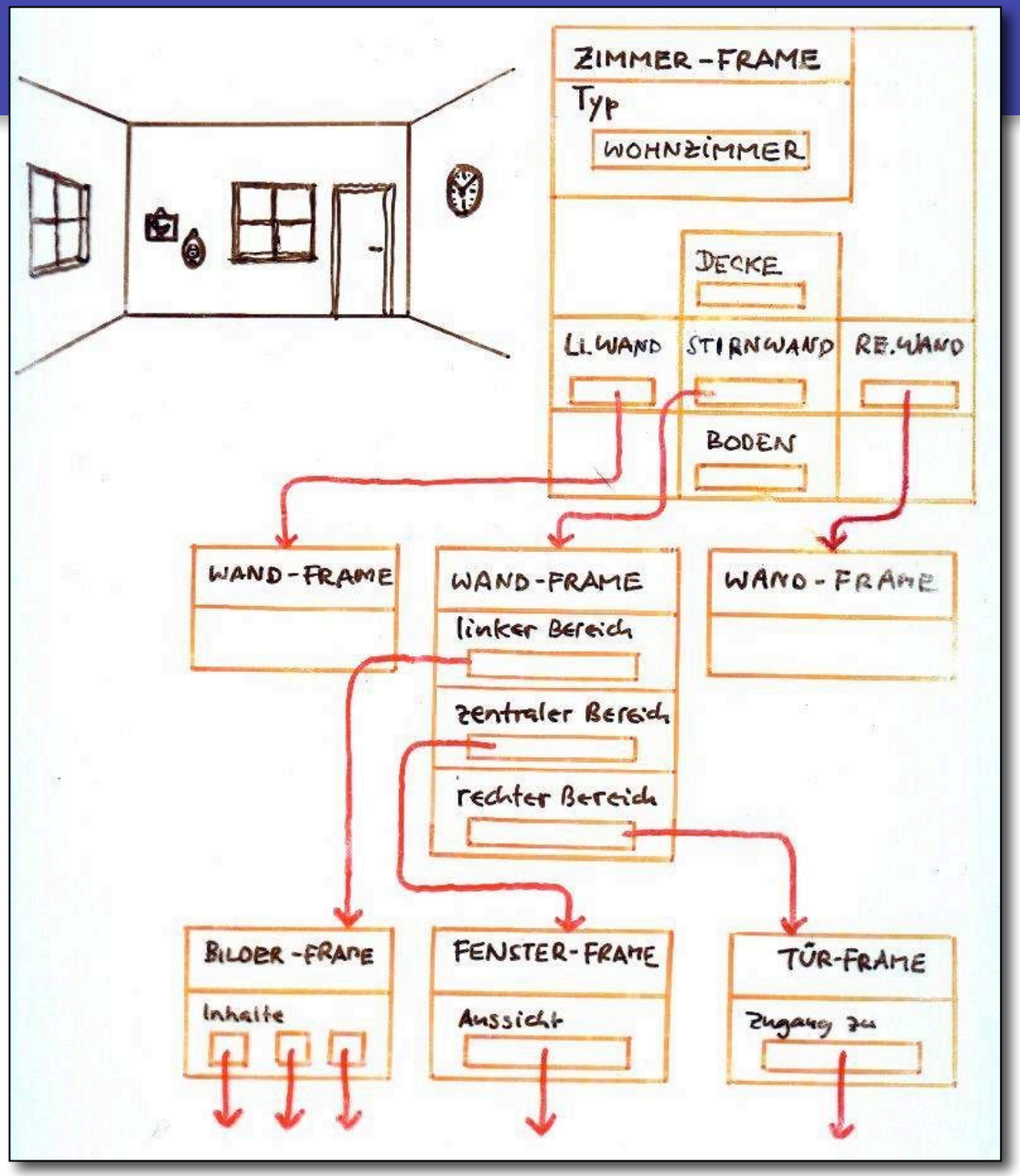


(Ausschnitt)

Objekt	Eigenschaften	Werte
Säugetier :	ist_ein :	Wirbeltier
	Vermehrung :	lebendgebärend
	hat :	Kopf, Beine
	...	...

Objekt	Eigenschaften	Werte
Elefant :	ist_ein :	Säugetier
	Farbe :	grau
	hat :	Rüssel
	Größe :	groß
	Lebensraum :	Boden

Objekt	Eigenschaften	Werte
Clyde :	instanz_von :	Elefant
	Farbe :	grau
	hat :	Rüssel
	Größe :	groß
	Lebensraum :	Boden


ZIMMER-FRAME
Typ
WOHNZIMMER
DECKE
LI.WAND
STIRNWAND
RE.WAND
BODEN
WAND-FRAME
WAND-FRAME
linker Bereich
zentraler Bereich
rechter Bereich
WAND-FRAME
BILDER-FRAME
Inhalte
FENSTER-FRAME
Aussicht
TÜR-FRAME
Zugang zu

Ursprung von Frames

Marvin Minsky (1975): A framework for representing knowledge. In P.H. Winston (ed.): The Psychology of Computer Vision. New York: McGraw-Hill.

Kognitive Theorie über:

- Wiedererkennen von stereotypen Objekten (z.B. Wohnzimmer)
- Handeln bei stereotypen Ereignissen (z.B. Kindergeburtstag)
- Beantwortung von Fragen über stereotype und konkrete Objekte

Leseempfehlung z.B.:

- Charniak & McDermott, Kapitel 1, Seite 11-29