Universität Bielefeld

Artificial Intelligence Methods

Marc Erich Latoschik
AI & VR Lab
Artificial Intelligence Group
University of Bielefeld

Universität Bielefeld

Knowledge Representation

*parts from (Russel & Norvig, 2004) Chapter 10

Outline

- Internal and symbolic representation
- Sentence structure
- Ontological engineering
- Categories and objects
- Actions, situations and events
- Mental events and mental objects
- The internet shopping world
- Reasoning systems for categories
- Reasoning with default information
- Truth maintenance systems

Internal Representation

central for „reasoning“:
internal representation
and
symbol manipulation.

- Representation in general:
  An idealized world description
  (not necesserily symbolic)
- Internal symbolic representation:
  requires a common symbol language, in which an agent can express and manipulate propositions about the world.
- good choice
  for symbolic representations are languages of logic, however, some preparations have to be made...

Make References Explicit

Natural language often is ambiguous:

The chair was placed on the table.
It was broken.

The chair (r1) was placed on the table (r2).
It (r1) was broken.

(Now it becomes obvious what was broken.)

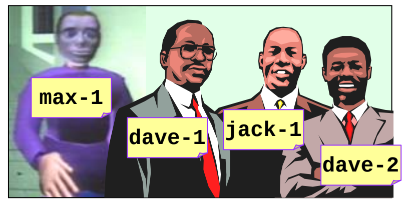
Der Hund (r1) saß auf dem Tisch (r2).
Et (r1) bellte

(Now it becomes obvious what barks.)

The same name can be used multiple times:

Dave should do it!

„Dave“ who?



Refrential Uniqueness

☆ **1. Postulation: Symbolic representations must explictely define relations for entity references!**

i.e., all ambiguity with respect to entities must be eliminated in the internal representation:

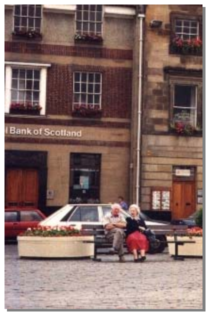
- **all individuals get a unique name**
- **this means only one individual per name**

Hence: instead of multiple "Daves": `dave-1, dave-2` etc.

Such unique names are denoted as *instances* or *tokens*.

Semantic Uniqueness

☆ **2. Postulation: All symbols of an internal representation must be unique ("unambiguous")!**



Examples for semantic („word-sense"-) ambiguity:

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]**

Different symbols imply different semantics
(even if their linguistic roots might be the same):

For example, who caught a cold must sneeze.

Functional Uniqueness

☆ **3. Postulation: Internal representations must uniquely express the functional roles!**

Petra catches the ball.
The ball Petra catches.
The ball is caught by Petra.

Who is the catcher? **Who or what is the caught object?**

Conclusion: **Symbolic representations must be unique regarding several aspects:**

- **referential**
- **semantic**
- **functional**

From Linguistic Sentence to Representation

simple example

Jack caught a ball.

↓ **disambiguate references**

`jack-2` caught `ball-5.`

↓ **disambiguate word sense**

`jack-2 catch-object ball-5 .`

↓ **brackets as delimiter**

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

↓ **predicate as prefix**

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

remove/add syntactic sugar

Predicates, Logic Sentences, Assertions

For the linguistical catch, we introduce a (2-ary) predicate `catch-object` in the representation:

```
catch-object(Jack-2, Ball-5)
(catch-object jack-2 ball-5)
```

predicate (→ `catch-object`), arguments (→ `jack-2`, `ball-5`)

p(A,B) ⟶ (p a b)

syntactic sugar

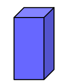
A logic sentence defines a fact about one or multiple entities, in this case a catch relation between one Jack and a specific ball.

- **Assertions are logic sentences which we take as given facts (as elements of an actual internal representation)**

Linguistic Sentence and Representation

In general, a linguistig sentence is represented by <u>multiple</u> logic sentences:

Jack caught a blue block.




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

Processes operating on internal representations are used to deduct derive new facts from known facts: *Inference*

Commonly used inference concept and term: *Deduction*

Such processes can be modeled in higher order logic. (Normally we use first order logic.)

Slot-Assertion-Notation

Reason: To express functional relations

These are still FOL representations but they express more (using the slot-predicates): *Functional structure*

Examples:

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

predicate — arguments (slots)

are representes as: **(still in FOL)**

```
(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-predicates

Slot-and-Filler-Notation (–>Frames)

Different slot-assertions are combined to provide a structured expression

A set of facts (assertions) becomes an object-centered format.

Object in this case:

The „catch-object"-event `catch-22`

this:

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

becomes:

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

general structure:

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

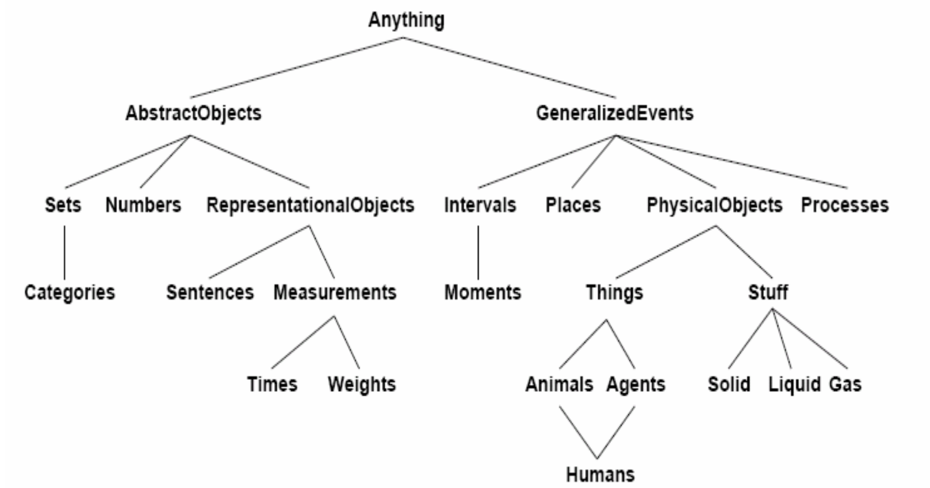
(object centered format!)

slot ↑ filler ↑

Ontological engineering

- How to create more general and flexible representations.
  - Concepts like actions, time, physical object and beliefs
  - Operates on a bigger scale than K.E.
- Define general framework of concepts
  - Upper ontology
- Limitations of logic representation
  - Red, green and yellow tomatoes: exceptions and uncertainty

The upper ontology of the world



Difference with special-purpose ontologies

- A general-purpose ontology should be applicable in more or less any special-purpose domain.
  - Add domain-specific axioms
- In any sufficiently demanding domain different areas of knowledge need to be unified.
  - Reasoning and problem solving could involve several areas simultaneously
- What do we need to express?

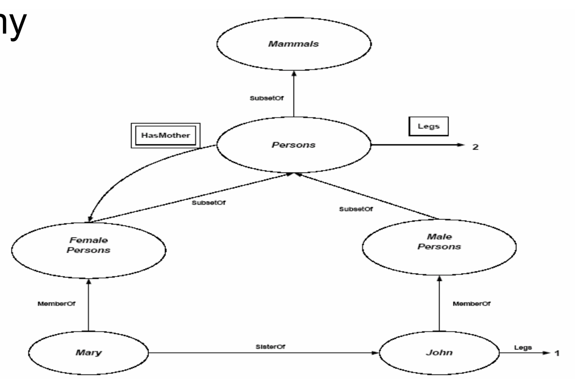
  Categories, Measures, Composite objects, Time, Space, Change, Events, Processes, Physical Objects, Substances, Mental Objects, Beliefs

Categories and objects

- KR requires the organisation of objects into **categories**
  - Interaction at the level of the object
  - Reasoning at the level of categories
- Categories play a role in predictions about objects
  - Based on perceived properties
- Categories can be represented in two ways by FOL
  - Predicates: apple(x)
  - *Reification* of categories into objects: apples
- Category = set of its members

Category organization

- Relation = *inheritance*:
  - All instance of food are edible, fruit is a subclass of food and apples is a subclass of fruit then an apple is edible.
- Defines a taxonomy



FOL and categories

- An object is a member of a category
  - MemberOf(BB_{12},Basketballs)
- A category is a subclass of another category
  - SubsetOf(Basketballs,Balls)
- All members of a category have some properties
  - $\forall$ x (MemberOf(x,Basketballs) $\Rightarrow$ Round(x))
- All members of a category can be recognized by some properties
  - $\forall$ x (Orange(x) $\wedge$ Round(x) $\wedge$ Diameter(x)=9.5in $\wedge$ MemberOf(x,Balls) $\Rightarrow$ MemberOf(x,BasketBalls))
- A category as a whole has some properties
  - MemberOf(Dogs,DomesticatedSpecies)

Relations between categories

- Two or more categories are ***disjoint*** if they have no members in common:
  - Disjoint(s)$\Leftrightarrow$($\forall$ c_1,c_2 $c_1 \in s \wedge c_2 \in s \wedge c_1 \neq c_2 \Rightarrow$ Intersection(c_1,c_2) ={})
  - Example:
    Disjoint({animals, vegetables})

- A set of categories *s* constitutes an ***exhaustive decomposition*** of a category *c* if all members of the set *c* are covered by categories in *s*:
  - E.D.(s,c) $\Leftrightarrow$ ($\forall$ i i $\in$ c $\Rightarrow \exists$ c_2 $c_2 \in$ s $\wedge$ i $\in$ c_2)
  - Example:

  ExhaustiveDecomposition({Americans, Canadian, Mexicans}, NorthAmericans).

Relations between categories

- A ***partition*** is a disjoint exhaustive decomposition:
  - $\text{Partition}(s,c) \Leftrightarrow \text{Disjoint}(s) \wedge \text{E.D.}(s,c)$
  - Example: Partition({Males,Females},Persons).
- Is ({Americans,Canadian, Mexicans},NorthAmericans) a partition?
  - No! There might be dual citizenships.
- Categories can be defined by providing necessary and sufficient conditions for membership
  - $\forall x\ \text{Bachelor}(x) \Leftrightarrow \text{Male}(x) \wedge \text{Adult}(x) \wedge \text{Unmarried}(x)$

Natural kinds

- Many categories have no clear-cut definitions (e.g., chair, bush, book).
- Tomatoes: sometimes green, red, yellow, black. Mostly round.
- One solution: subclass using category *Typical(Tomatoes).*
  - $\text{Typical}(c) \subseteq c$
  - $\forall x, x \in \text{Typical}(\text{Tomatoes}) \Rightarrow \text{Red}(x) \wedge \text{Spherical}(x).$
  - We can write down useful facts about categories without providing exact definitions.
- Wittgenstein (1953) gives an exhaustive summary about the problems involved when exact definitions for natural kinds are required in his book “Philosophische Untersuchungen”.
- What about “bachelor”? Quine (1953) challenged the utility of the notion of *strict definition*. We might question a statement such as “the Pope is a bachelor”.

Artificial Intelligence Methods – WS 2005/2006 – Marc Erich Latoschik

Physical composition

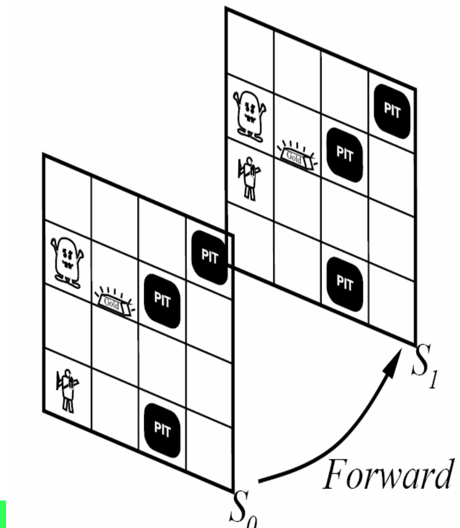
- One object may be part of another:
  - PartOf(Bucharest,Romania)
  - PartOf(Romania,EasternEurope)
  - PartOf(EasternEurope,Europe)
- The PartOf predicate is transitive (and reflexive), so we can infer that PartOf(Bucharest,Europe)
- More generally:
  - $\forall$ x PartOf(x,x)
  - $\forall$ x,y,z PartOf(x,y) $\wedge$ PartOf(y,z) $\Rightarrow$ PartOf(x,z)
- Often characterized by structural relations among parts.
  - E.g. Biped(a) $\Rightarrow$

$$(\exists l_1, l_2, b)(Leg(l_1) \wedge Leg(l_2) \wedge Body(b) \wedge$$
$$PartOf(l_1, a) \wedge PartOf(l_2, a) \wedge PartOf(b, a) \wedge$$
$$Attached(l_1, b) \wedge Attached(l_2, b) \wedge$$
$$l_1 \neq l_2 \wedge (\forall l_3)(Leg(l_3) \Rightarrow (l_3 = l_1 \vee l_3 = l_2)))$$

Measurements

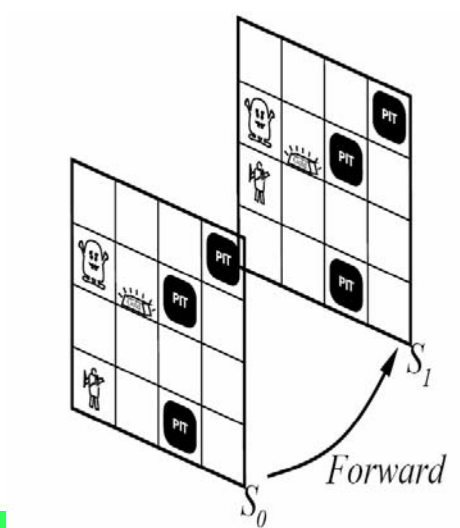
- Objects have height, mass, cost, ...
  Values that we assign to these are **measures**
- Combine Unit functions with a number:
  Length(L_1) = Inches(1.5) = Centimeters(3.81).
- Conversion between units:
  $\forall$ i Centimeters(2.54 x i)=Inches(i).
- Some measures have no scale:
  Beauty, Difficulty, etc.
  - Most important aspect of measures: they are **orderable**.
  - Don't care about the actual numbers. (An apple can have deliciousness .9 or .1.)

Actions, events and situations



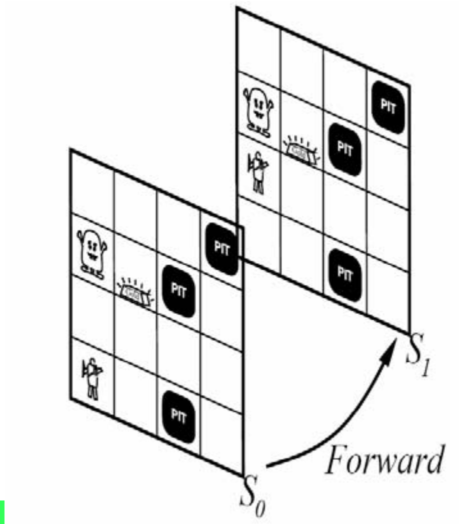
- Reasoning about outcome of actions is central to KB-agent.
- How can we keep track of location in FOL?
  - Remember the multiple copies in PL.
- Representing time by situations (states resulting from the execution of actions).
  - Situation calculus

Actions, events and situations



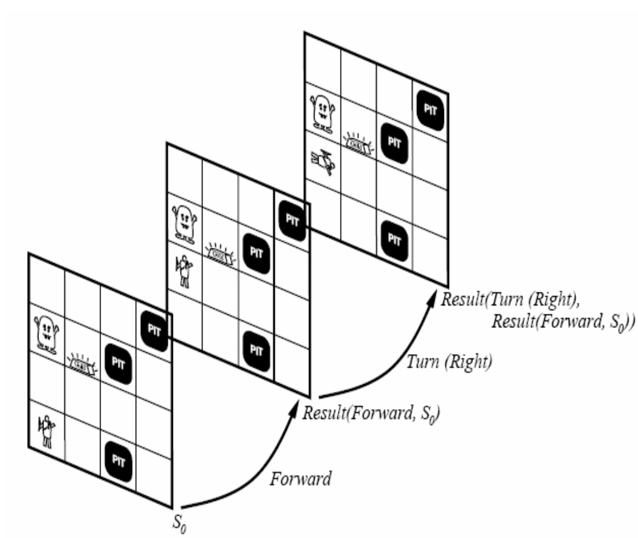
- Situation calculus:
  - Actions are logical terms
  - Situations are logical terms consiting of
    - The initial situation I
    - All situations resulting from the action on I (=*Result(a,s)*)
  - **Fluents** are functions and predicates that vary from one situation to the next.
    - E.g. $\neg Holding(G_1, S_0)$
  - **Eternal predicates** are also allowed
    - E.g. $Gold(G_1)$

Actions, events and situations



- Results of action sequences are determined by the individual actions.
- *Projection task:* an SC agent should be able to deduce the outcome of a sequence of actions.
- *Planning task:* find a sequence that achieves a desirable effect

Actions, events and situations



Describing change

- Simples Situation calculus requires two axioms to describe change:
  - Possibility axiom: when is it possible to do the action

    $At(Agent,x,s) \wedge Adjacent(x,y) \Rightarrow Poss(Go(x,y),s)$
  - Effect axiom: describe changes due to action

    $Poss(Go(x,y),s) \Rightarrow At(Agent,y,Result(Go(x,y),s))$
- What stays the same?
  - Frame problem: how to represent all things that stay the same?
  - Frame axiom: describe non-changes due to actions

    $At(o,x,s) \wedge (o \neq Agent) \wedge \neg Holding(o,s) \Rightarrow At(o,x,Result(Go(y,z),s))$

Representational frame problem

- If there are F fluents and A actions then we need AF frame axioms to describe other objects are stationary unless they are held.
  - We write down the effect of each actions
- Solution; describe how each fluent changes over time
  - Successor-state axiom:

    $Pos(a,s) \Rightarrow (At(Agent,y,Result(a,s)) \Leftrightarrow (a = Go(x,y)) \vee$

    $(At(Agent,y,s) \wedge a \neq Go(y,z))$
  - Note that next state is completely specified by current state.
  - Each action effect is mentioned only once.

Other problems

- How to deal with secondary (implicit) effects?
  - If the agent is carrying the gold and the agent moves then the gold moves too.
  - Ramification problem
- How to decide EFFICIENTLY whether fluents hold in the future?
  - Inferential frame problem.
- Extensions:
  - Event calculus (when actions have a duration)
  - Process categories

Mental events and objects

- So far, KB agents can have beliefs and deduce new beliefs
- What about knowledge about beliefs? What about knowledge about the inference proces?
  - Requires a model of the mental objects in someone's head and the processes that manipulate these objects.
- Relationships between agents and mental objects: believes, knows, wants, …
  - Believes(Lois,Flies(Superman)) with Flies(Superman) being a function … a candidate for a mental object (reification).
  - Agent can now reason about the beliefs of agents.

The internet shopping world

- A Knowledge Engineering example
- An agent that helps a buyer to find product offers on the internet.
  - IN = product description (precise or ¬precise)
  - OUT = list of webpages that offer the product for sale.
- Environment = WWW
- Percepts = web pages (character strings)
  - Extracting useful information required.

The internet shopping world

- Find relevant product offers

  RelevantOffer(page,url,query) ⇔ *Relevant(page, url, query)* ∧ *Offer(page)*

  - Write axioms to define Offer(x)
  - Find relevant pages: Relevant(x,y,z) ?
    - Start from an initial set of stores.
    - What is a relevant category?
    - What are relevant connected pages?
  - Require rich category vocabulary.
    - Synonymy and ambiguity
  - How to retrieve pages: *GetPage(url)*?
    - Procedural attachment
- Compare offers (information extraction).

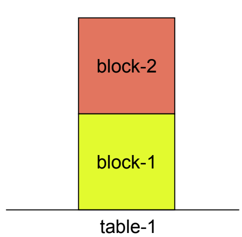
Reasoning systems for categories

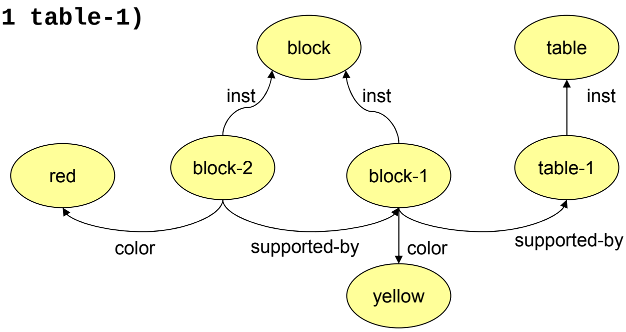
- How to organize and reason with categories?
  - Semantic networks
    - Visualize knowledge base
    - Efficient algorithms for category membership inference
  - Description logics
    - Formal language for constructing and combining category definitions
    - Efficient algorithms to decide subset and superset relationships between categories.

Representation of a Scene

```
(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)
```

- as set of logic expressions
- as Semantic Net





Semantic Networks

- Logic vs. semantic networks
- Many variations
  - All represent individual objects, categories of objects and relationships among objects.
- Allows for inheritance reasoning
  - Female persons inherit all properties from person.
  - Cfr. OO programming.
- Inference of inverse links
  - SisterOf vs. HasSister

Alternative Notations

Semantic Nets (a.k.a. „associative nets) and FOL sentences represent same information in different formats:

Nodes correspond to terms
marked out directed edges correspond to predicates

➢ **they are alternative notations for the same content, not in principle different representations!**

☆ **What differs?**

Missing existential quantifier Functions (extensions exist)
Semantic nets additionally provide pointers (and sometimes back pointers) which allow easy and high-performance information access (e.g., to instances): INDEXING

[similar: frames]

ISA-Hierarchy and Inheritance

isa:
"is a"
"ist ein"

- Key concept in the tradition of semantic nets
- Instances inherit properties which we attribute to sets of individuals (classes).
- This can be propagates along the complete isa hierarchy
  - Inheritance of properties
  - Reason: Knowledge representation economy

inst:
"instance of"
„Instanz von"

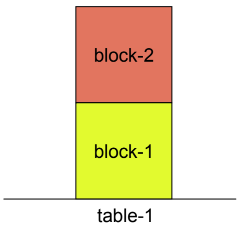
- Search along isa- and inst-links to access information not directly associated (using inheritance)
  - inst: $\in$ **member of**
  - isa: $\subseteq$ **subset of**

Semantic networks

- Drawbacks
  - Links can only assert binary relations
  - Can be resolved by reification of the proposition as an event
- Representation of default values
  - Enforced by the inheritance mechanism.

Representation of a Scene

```
(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 notations"

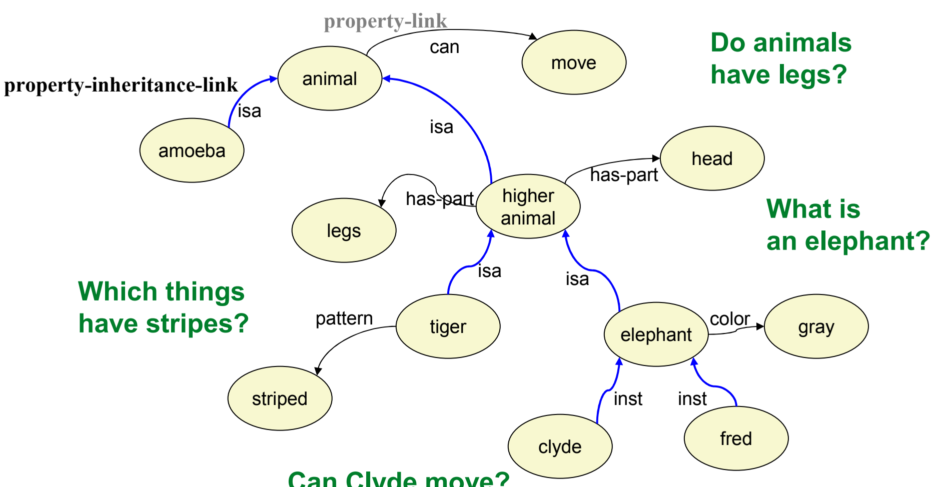
• as frames (slot-and-filler-Notation)

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

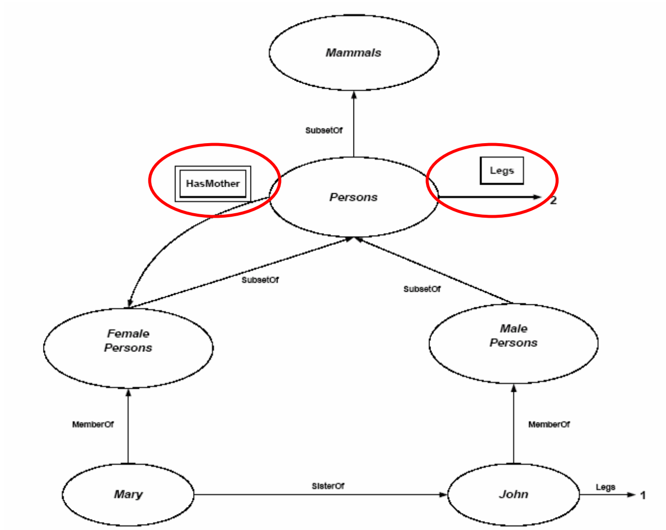
Frame	Attribute (slots)	Werte (fillers)
block-1 :	inst :	block
	color :	yellow
	supported-by :	table-1
	...	...

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

Example of an ISA-Hierarchy



Semantic network example



Universal vs. individual Properties

NOTE – distinguish:

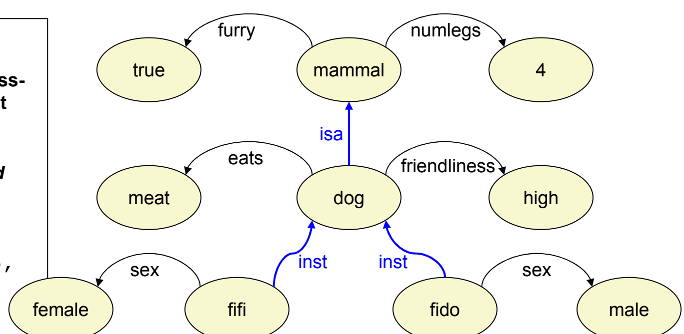
1. **Property links from class-nodes in a semantic net `(dog, mammal)`:**

implict universally quantified assertions *

2. **Property links from instance-nodes `(fido, fiffy)`:**

assertions for individuals
e.g., `(sex fifi female)`

Type vs. token!

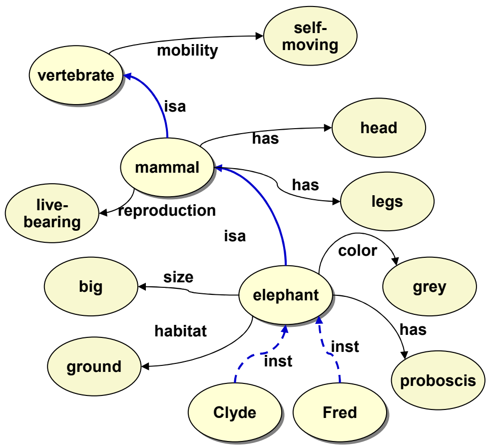


*example: dog-property reconstruction in FOL:

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

Inheritance in Semantic Nets and Frames

(slightly different modelling than before)

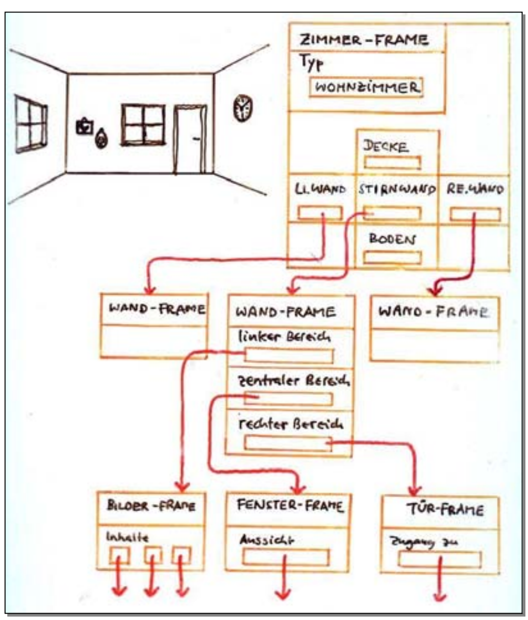


(fragment)

object	property	value
mammal :	isa :	vertebrate
	reproduction :	livebearing
	has :	head, legs
	...	...

object	property	value
elephant :	isa :	mammal
	color :	grey
	has :	proboscis
	size :	big
	habitat :	Boden

object	property	value
Clyde :	inst :	elephant
	color :	grey
	has :	proboscis
	size :	big
	habitat :	ground



Origin of Frames

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

Cognitive theory about:

- Recognition of stereotype objects (e.g., living room)
- Action for stereotype events (e.g., children‘s birthday party)
- Replying to questions about stereotype or specific objects.

recommended reading, e.g.:

- Charniak & McDermott, chapter 1, pages 11-29

Description logics

- Are designed to describe defintions and properties about categories
  - A formalization of semantic networks
- Principal inference task is
  - *Subsumption*: checking if one category is the subset of another by comparing their definitions
  - *Classification*: checking whether an object belongs to a category.
  - *Consistency*: whether the category membership criteria are logically satisfiable.

Reasoning with Default Information

- "The following courses are offered: CS101, CS102, CS106, EE101"
  - Four (db)
    - Assume that this information is complete (not asserted ground atomic sentences are false)

    = CLOSED WORLD ASSUMPTION
    - Assume that distinct names refer to distinct objects

    = UNIQUE NAMES ASSUMPTION
  - Between one and infinity (logic)
    - Does not make these assumptions
    - Requires completion.

Truth maintenance systems

- Many of the inferences have default status rather than being absolutely certain
  - Inferred facts can be wrong and need to be retracted = BELIEF REVISION.
  - Assume KB contains sentence P and we want to execute TELL(KB, $\neg$P)
    - To avoid contradiction: RETRACT(KB,P)
    - But what about sentences inferred from P?
- Truth maintenance systems are designed to handle these complications.