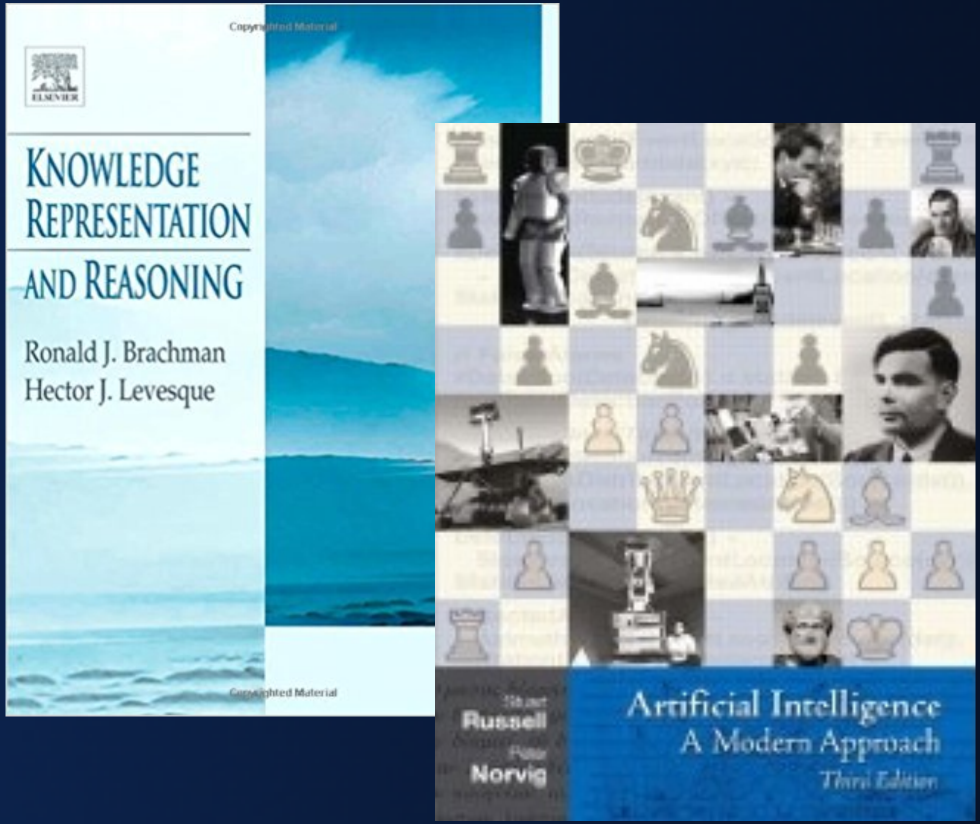


AI Fundamentals: Knowledge Representation and Reasoning

Maria Simi

IN SUPREMÆ DIGNITATIS
1343

Graph representations and structured representations

LESSON 5: SEMANTIC NETWORKS, FRAMES

Summary

- ✓ Structured representations and graph representations
- ✓ Early semantic networks
- ✓ Sowa conceptual graphs
- ✓ KL-One
- ✓ Knowledge graphs
- ✓ Formalizing inheritance
- ✓ Frames and frame languages

Psychological-linguistic approach to KR&R

- The **logical approach**: devised to model rational reasoning
  - Emphasis on valid reasoning and formalization of mathematical properties
  - Extended to model commonsense reasoning
- The **cognitive–linguistic** approach
  - More concerned with understanding the mechanisms for knowledge acquisition, representation and use of knowledge in human minds
- Synergies with other fields
  - Cognitive psychology
  - Linguistics

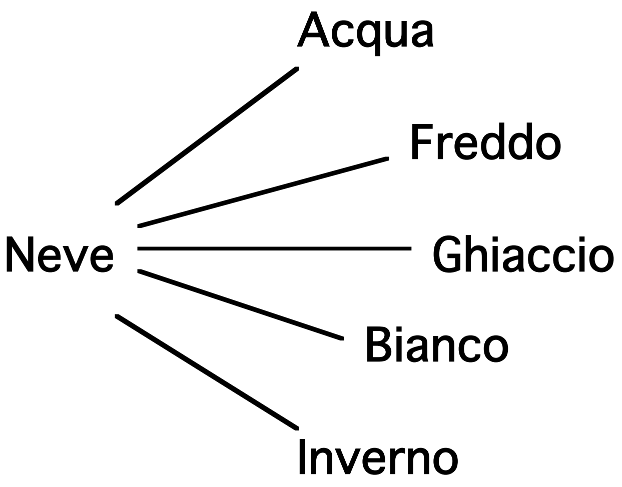
Associationist theories

In logic systems, symbolic expressions are syntactically transformed without paying attention to the symbols used or on their interpretation; the symbol themselves are arbitrary.

$$\forall x\ \text{Strawberry}(x) \Rightarrow \text{Red}(x)$$

Associationist theories are instead concerned with the connections among symbols and the meaning that emerges from these connections.

The idea is that meaning of a word emerges as a result of the connections to other words.



Neve

Acqua

Freddo

Ghiaccio

Bianco

Inverno

Semantic memory

The question is how the **meaning of words** is acquired, represented and used.

The memory itself is distinguished in:

- *Episodic memory*: specific facts and events
- *Semantic memory*: abstract and general knowledge

Semantic networks is a graphical model proposed for *semantic memory*

Two kinds of knowledge:

- Concepts: the semantic counterpart of words, represented as nodes
- Propositions: relations among concepts, represented as labelled arcs

Not accounting for dynamic aspects of memory and learning: Other models:

- Distributional models
- Connectionist models

Hierarchical organization of concepts: experiments

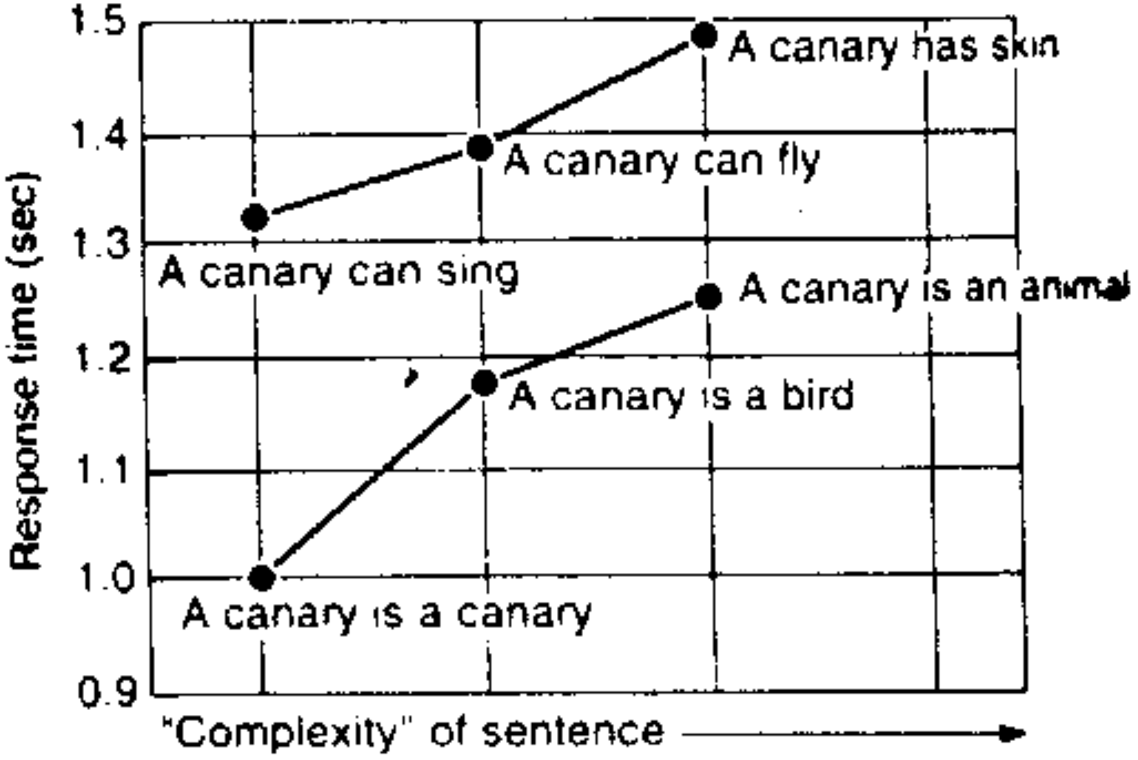
Cognitive psychology is an experimental discipline. Experiments by Collins & Quillian, 1969.

Questions:

1. "*Is a canary a bird*?"
2. "*<u>Does a canary fly</u>*?"
3. "*Does a canary has skin*?"

Answer times:

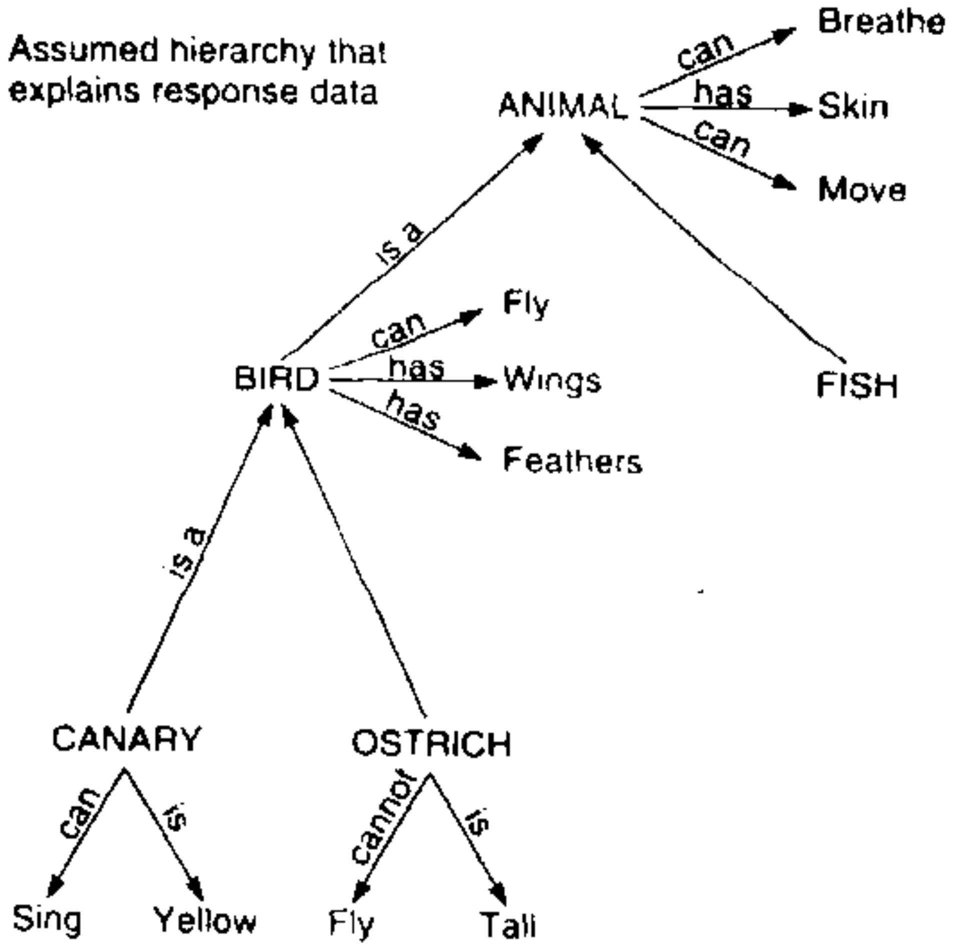
$T_1 < T_2 < T_3$



Hierarchical organization of concepts: interpretation

1. Evidence for hierarchical organization.
2. Properties are attached to the most general concept to which they apply.
3. Exceptions are attached directly to the object

Success of hierarchical organization of concepts in computer science and influence on OOP languages in SW engineering.


Assumed hierarchy that explains response data

The essence of semantic networks

Semantic networks are a large family of graphical representation schemes.

A semantic network is a graph where:

- Nodes, with a label, correspond to **concepts**
- Arcs, labelled and directed, correspond to binary relations between concepts, often called **roles**.

Nodes come in two flavors:

1. **Generic concepts**, corresponding to categories/classes
2. **Individual concepts**, corresponding to individuals

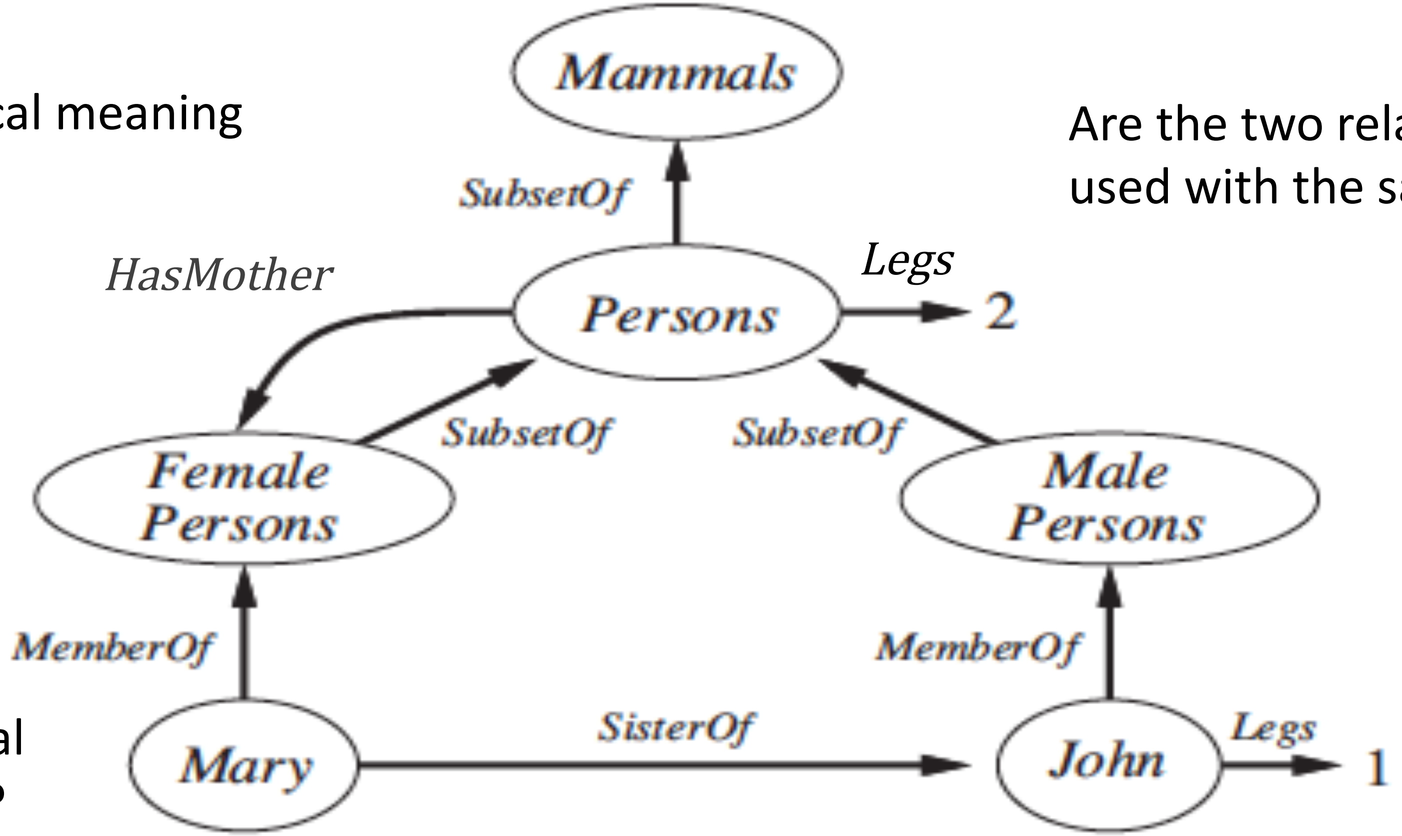
Two special relations are always present, with different names:

1. **IS-A**: holds between two generic concepts (subclass)
2. **Inst-Of**: holds between an individual concept and a class (member of)

An example [AIMA]

Which is the logical meaning of *HasMother*?

Are the two relations *Legs* used with the same meaning?



Which is the logical meaning of *Mary*?

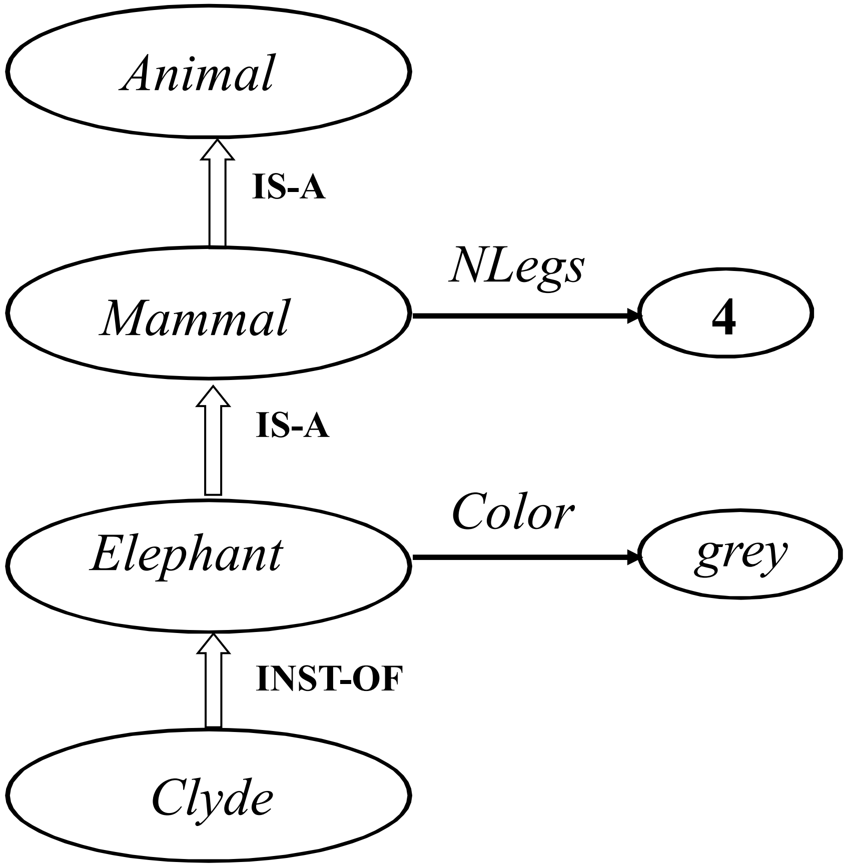
Inheritance in hierarchical networks

Inheritance is very conveniently implemented as **link traversal**.

How many legs has Clyde?

Just follow the INST-OF/IS-A chain until you find the property *NLegs*.

Multiple inheritance is also allowed, but is banned di OOP.

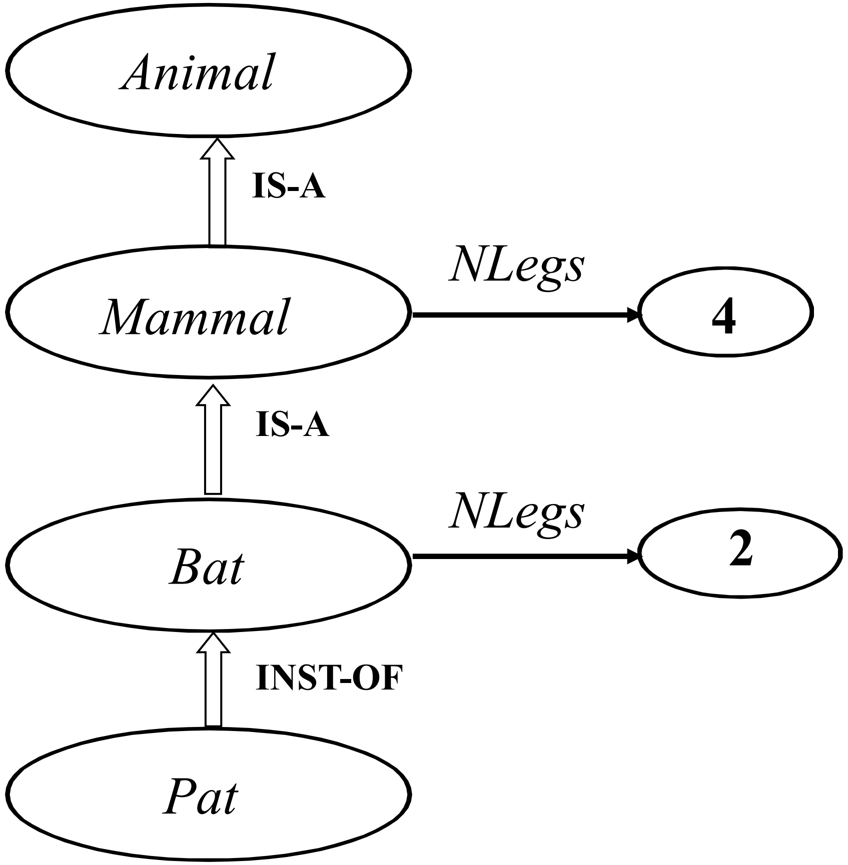


Inheritance with exceptions

The presence of exceptions does not create any problem.

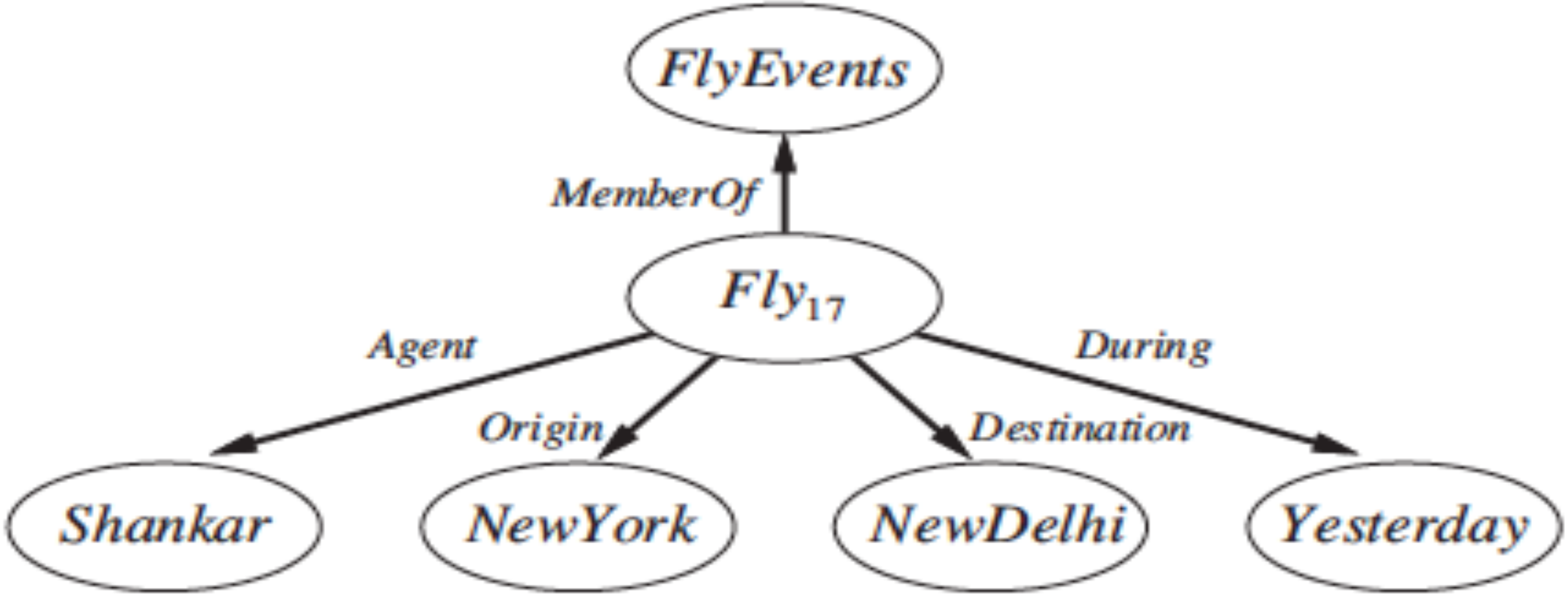
How many legs has Pat?

Just take the most specific information: the first that is found going up the hierarchy.



Only binary relations?

Case structure representation



Limited expressive power of semantic nets

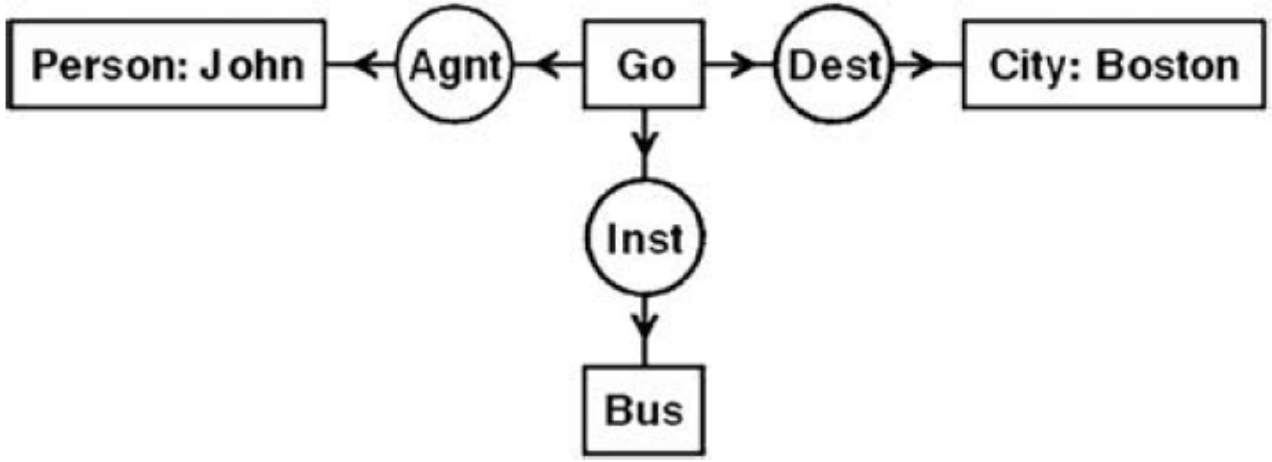
Even if they can express n-ary predicates, semantic networks do not have the same expressive power of FOL:

- Existentials, ∨, ⇒ ... are not expressible or are expressible only in special cases
- This is not necessarily a bad thing since it suggests a subset of FOL with interesting computational properties, explored in description logics.
- More expressive **assertional** networks were proposed in the past:

  Gottlob Frege (1879) developed a tree notation for the first version of first-order logic.

  Charles S. Peirce (1880, 1885) independently developed an algebraic notation for the modern notation for predicate calculus, that he called "The logic of the future".
- A well known example in AI are Sowa's **conceptual graphs**, inspired by Pierce's existential graphs. They candidate as an intermediate schema for representing natural language.

Sowa’s conceptual graphs



- Rectangles are concepts (possibly typed as in **Person: John**)
- Circles are called conceptual relations: the label is the type of the relation
- Logical meaning:

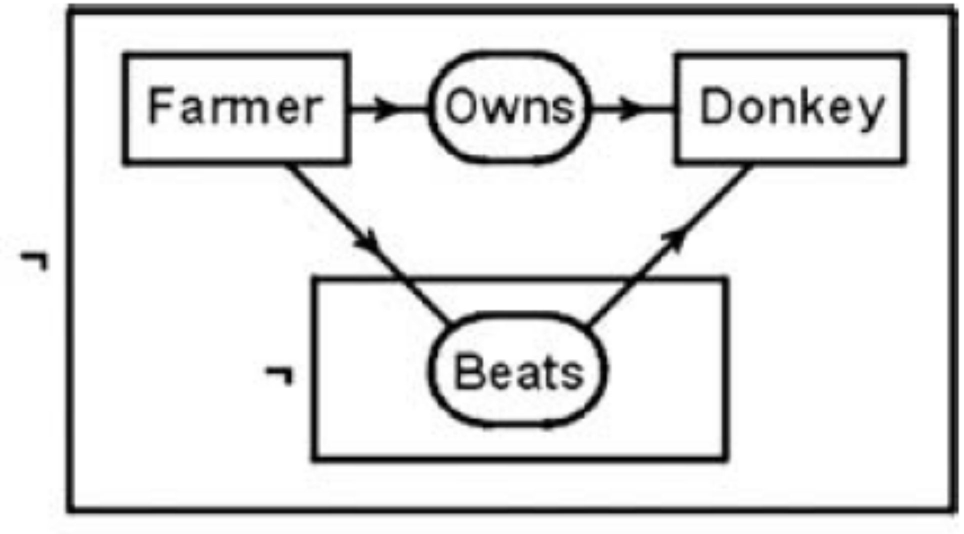
$$\exists x \, \exists y \, (Go(x) \wedge Person(John) \wedge City(Boston) \wedge Bus(y) \wedge$$
$$Agnt(x, John) \wedge Dest(x, Boston) \wedge Inst(x, y))$$

Sowa’s conceptual graphs

A more complex proposition, including implication, uses **context boxes**:

“If a farmer owns a donkey, then he beats it”

$$\neg[\exists x\, \exists y\, (Farmer(x) \wedge Donkey(y) \wedge$$
$$Owns(x, y) \wedge$$
$$\neg Beats(x, y))]$$



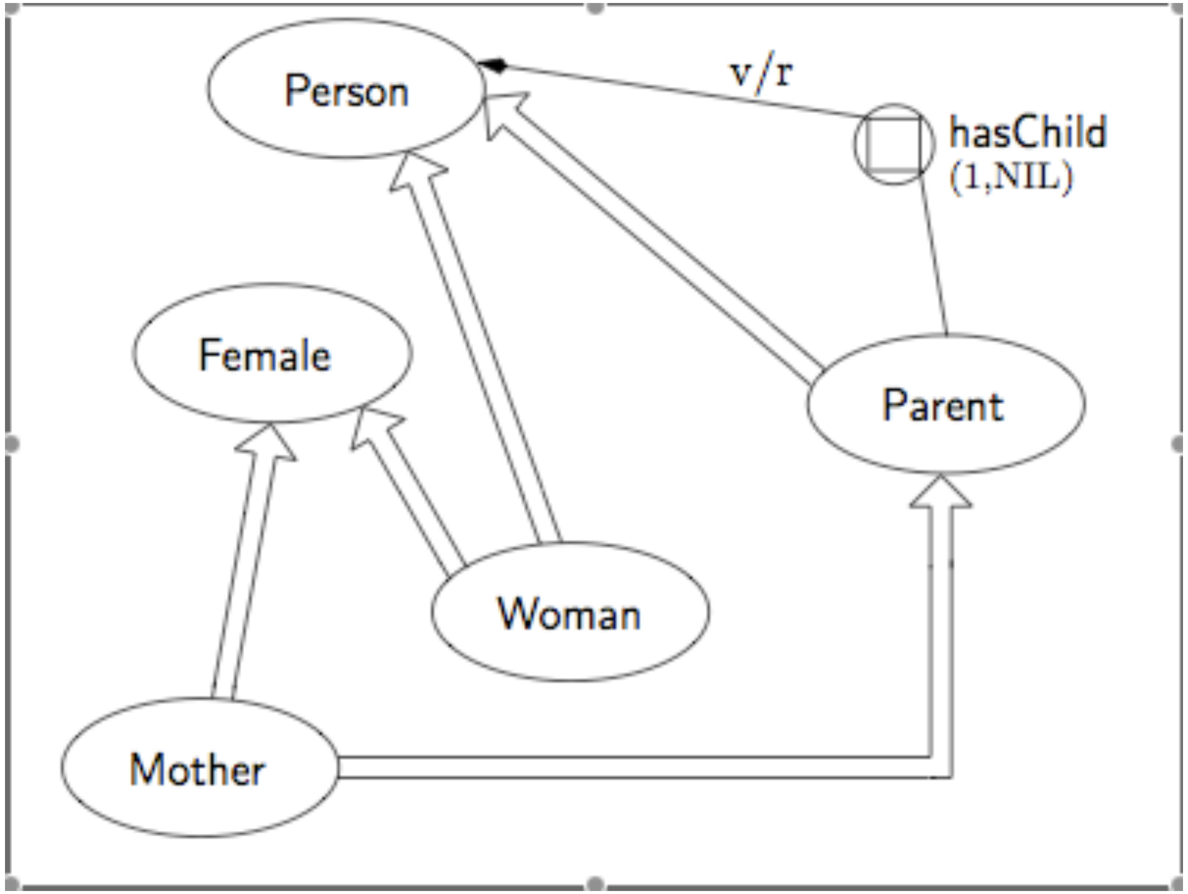
An example from KL-One

Woods [75] and others point out the lack of “semantics” of semantic nets

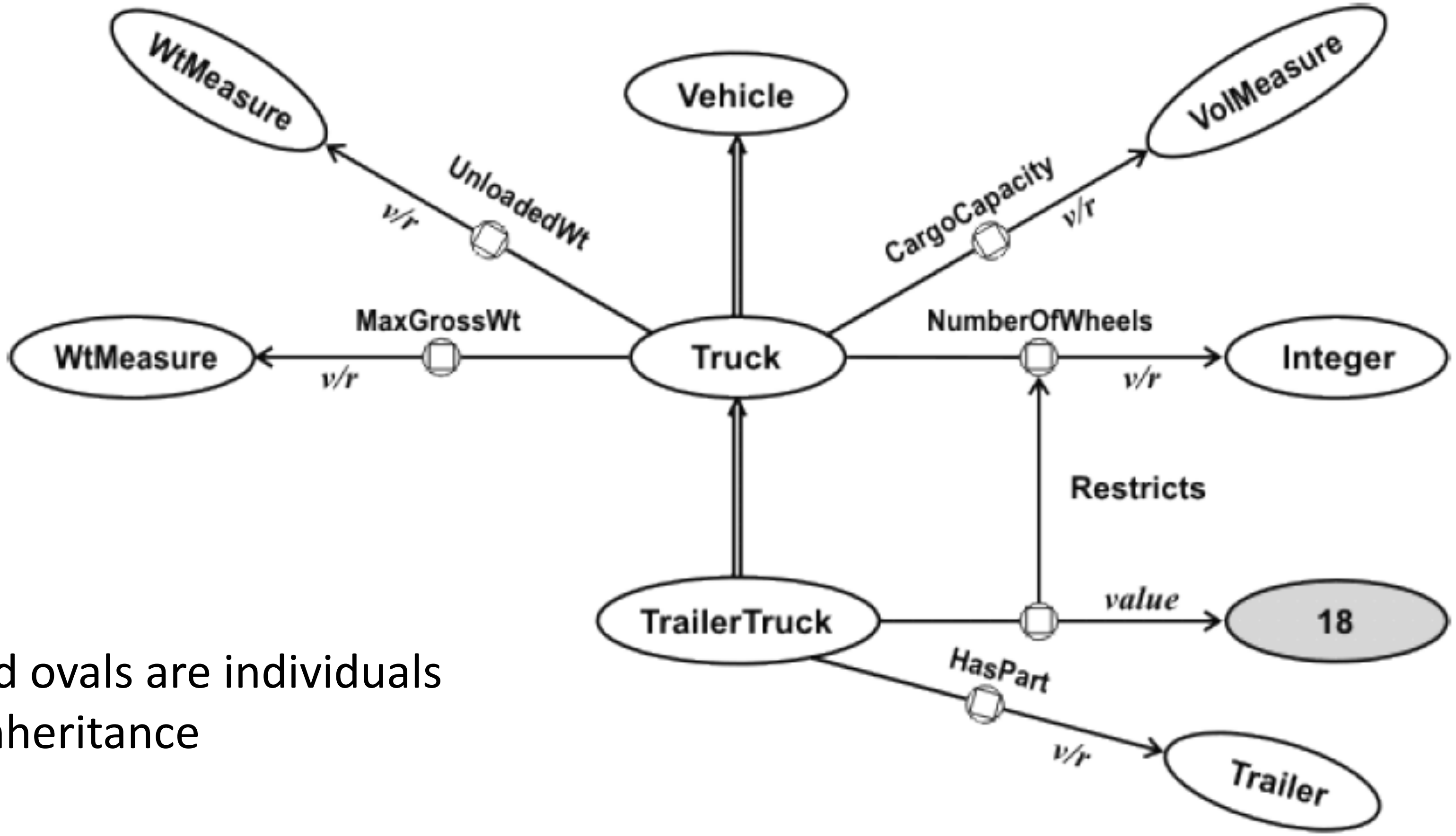
As a result in the 80’s KL-One [Brachman-Schmolze 1985] introduces important ideas:

- Concepts and roles (they are also nodes with a different status)
- Value restrictions (v/r)
- Numerical restrictions (1, NIL)
- **A formal semantics**

The double arrow is a IS-A relation.



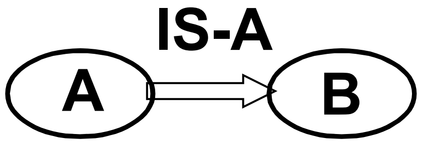
A more complex example in KL-One



- Shaded ovals are individuals
- Role inheritance

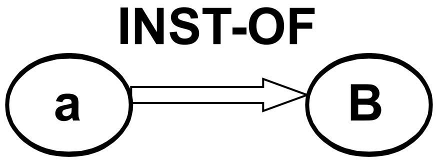
Logic account of semantic networks

We may look at semantic networks as a **convenient notation** for a part of FOL.
A representation and mechanism at the *symbol level* rather than at the *knowledge level.*



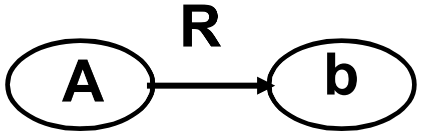
$\forall x\, A(x) \Rightarrow B(x)$

All members of class A are also member of class B.



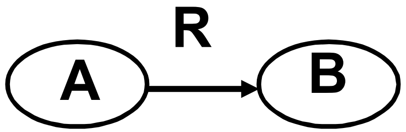
$B(a)$

a belongs to class B



$\forall x\;\; x \in A \Rightarrow R(x, b)$

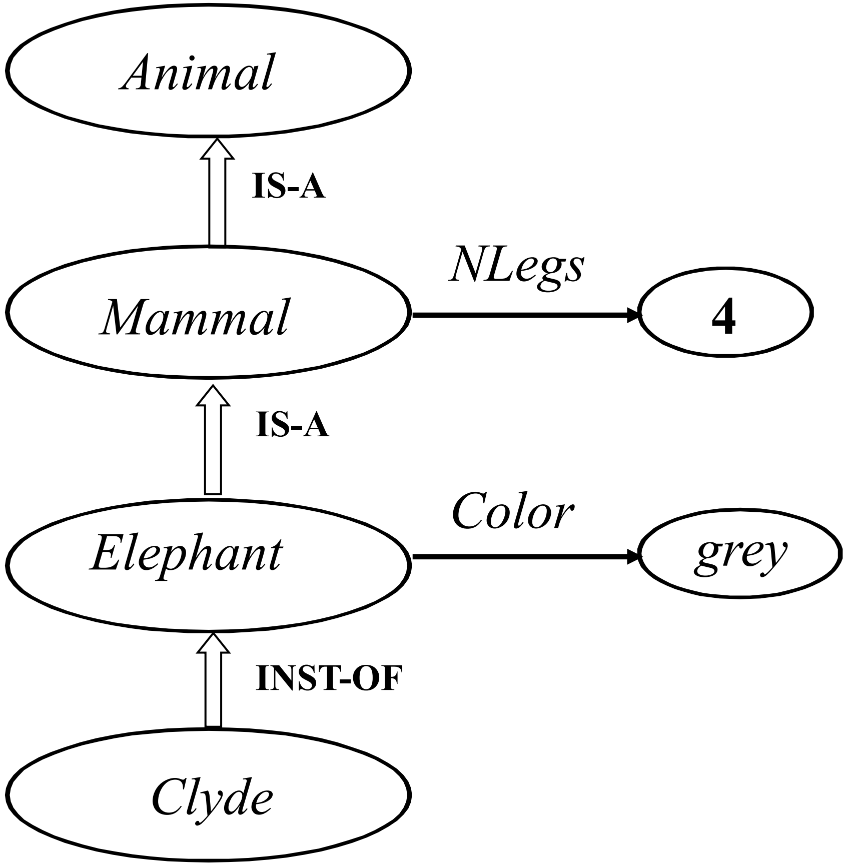
All members of class A are in relation R with b



$\forall x\;\; x \in A \Rightarrow \exists y\;\; y \in B \wedge R(x, y)$

For all members A, there is a R-related element in B

Translation example



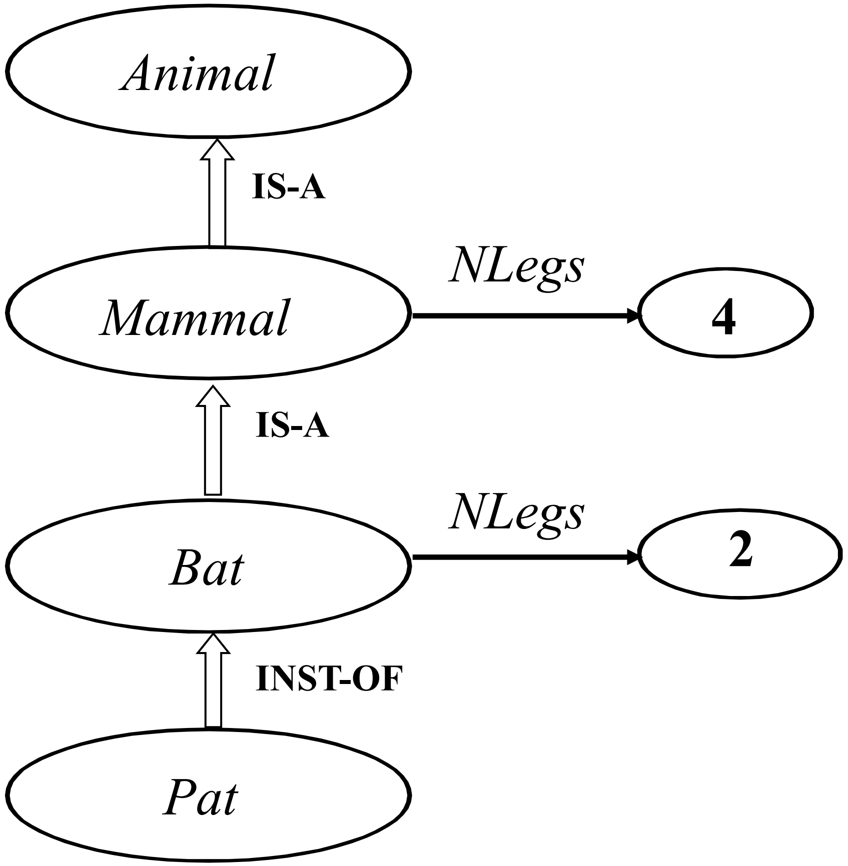
$\forall x\, Mammal(x) \Rightarrow Animal(x)$
$\forall x\, Mammal(x) \Rightarrow NLegs(x, 4)$
$\forall x\, Elephant(x) \Rightarrow Mammal(x)$
$\forall x\, Elephant(x) \Rightarrow Color(x, grey)$
$Elephant(Clyde)$

It is possible to deduce:

$Animal(Clyde)$
$Mammal(Clyde)$
$NLegs(Clyde, 4)$
$Color(Clyde, grey)$

Inheritance corresponds to $\forall$E, MP and transitivity of $\Rightarrow$

Accounting for exceptions



$\forall x\ Mammal(x) \Rightarrow Animal(x)$
$\forall x\ Mammal(x) \Rightarrow NLegs(x, 4)$
$\forall x\ Bat(x) \Rightarrow Mammal(x)$
$\forall x\ Bat(x) \Rightarrow NLegs(x, 4)$
$Bat(Pat)$

It is possible to deduce:

$NLegs(Pat, 4)$
$NLegs(Pat, 2)$

and this may lead to a contradiction.

Rewriting the rule as:

$\forall x\ Bat(x) \land x \neq Pat \Rightarrow NLegs(x, 4)$

Problem: logics are **flat**.

Defaults require nonmonotonic reasoning

Implementing defeasible inheritance

A more essential notation for inheritance networks: negated arcs

(a) More specific information should win

(b) Multiple inheritance. Case of ambiguous network.

Not so easy. Two strategies:

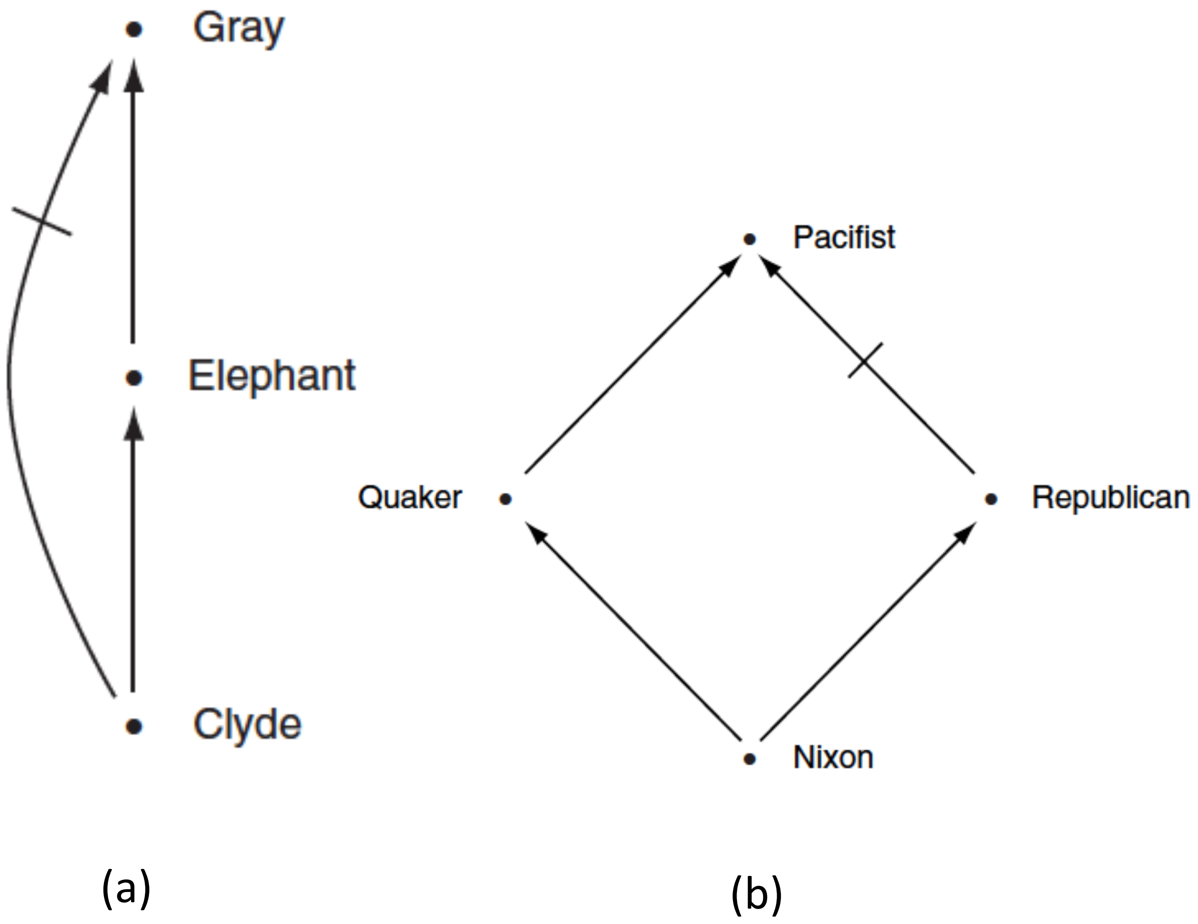
1. **Shortest path heuristics**

   it fails in presence of *redundant links* (shortcuts) -> (c) next page.

2. **Inferential distance**

   Based on the topology (not path length)

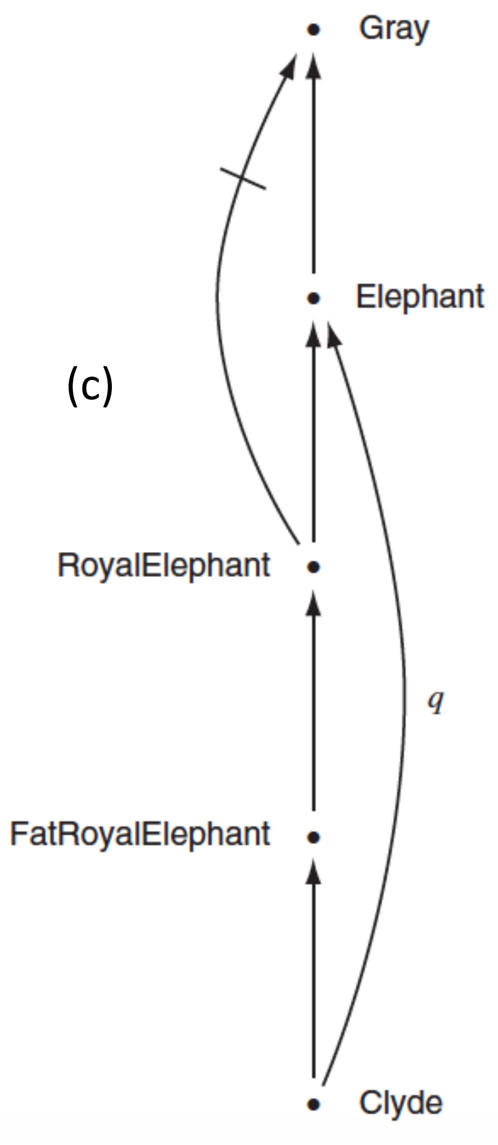
   A closer to B than C iff there is a path from A to C through B.



The *shortest distance* strategy would conclude that

Clyde is grey

But the information about **RoyalElephant** is more specific than **Elephan**t and should be preferred.



The *inferential distance* strategy would conclude that

Clyde is not grey

RoyalElephant is closer to Clyde than **Elephan**t because there is path to **Elephant** passing trough **Royal Elephant.**

A formal account of defeasible inheritance

Def. An inheritance hierarchy $\Gamma = \langle V, E \rangle$ is a directed, acyclic graph with positive and negative edges

- V are the nodes, or vertices
- E are the edges: Positive edges will be written as $(a \cdot x)$ and negative edges will be written as $(a \cdot \neg x)$.
- A **positive path** is a sequence of one or more positive edges $a \cdot \ldots \cdot x$ (supports $a \Rightarrow x$)
- A **negative path** is a sequence of zero or more positive edges followed by a single negative edge: $a \cdot \ldots \cdot v \cdot \neg x$ (supports $a \Rightarrow \neg x$)

A single conclusion can be supported by many paths.

However, not all arguments are equally believable. Only admissible paths are.

SKIPPED

Admissible paths

A path $a \cdot s_1 \cdot s_2 \ldots s_n \cdot [\neg]x$ is admissible *iff* every edge in it is **admissible with respect to *a*.**

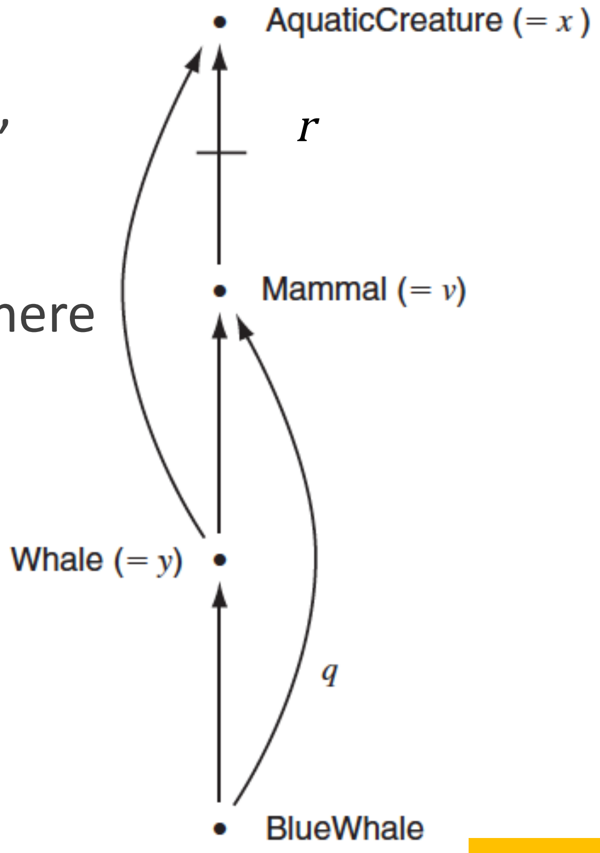
An edge is admissible with respect to *a* if there is a **non redundant**, admissible path leading to it from *a* that contains **no preempting** intermediaries.

Formally: An edge $v \cdot [\neg]x$ is admissible in Γ with respect to *a* if there is a positive path $a \cdot s_1 \cdot s_2 \ldots s_n \cdot v$ ($n \geq 0$) in E and

1. each edge in $a \cdot s_1 \cdot s_2 \ldots s_n \cdot v$ is admissible in Γ with respect to *a* (recursively)
2. no edge in $a \cdot s_1 \cdot s_2 \ldots s_n \cdot v$ is *redundant* in Γ with respect to *a*
3. no intermediate node *a*, s_1, s_2, … s_n is a *preemptor* of $v \cdot \neg x$ with respect to *a*

Redundant: if it is a sort of shortcut to the same conclusion (q in figure)

Preemptor: if defies the conclusion (Whale preempts negative edge r)



SKIPPED

WordNet [Miller]

A big lexical resource organized as a Semantic network (122.000 terms)

- names, verbs, adjectives, adverbs are organized in sets of synonims (synsets) which are a representation of a concepts (117.000 *synset*);
- A syntactic word is associated with a set of synsets: the word senses

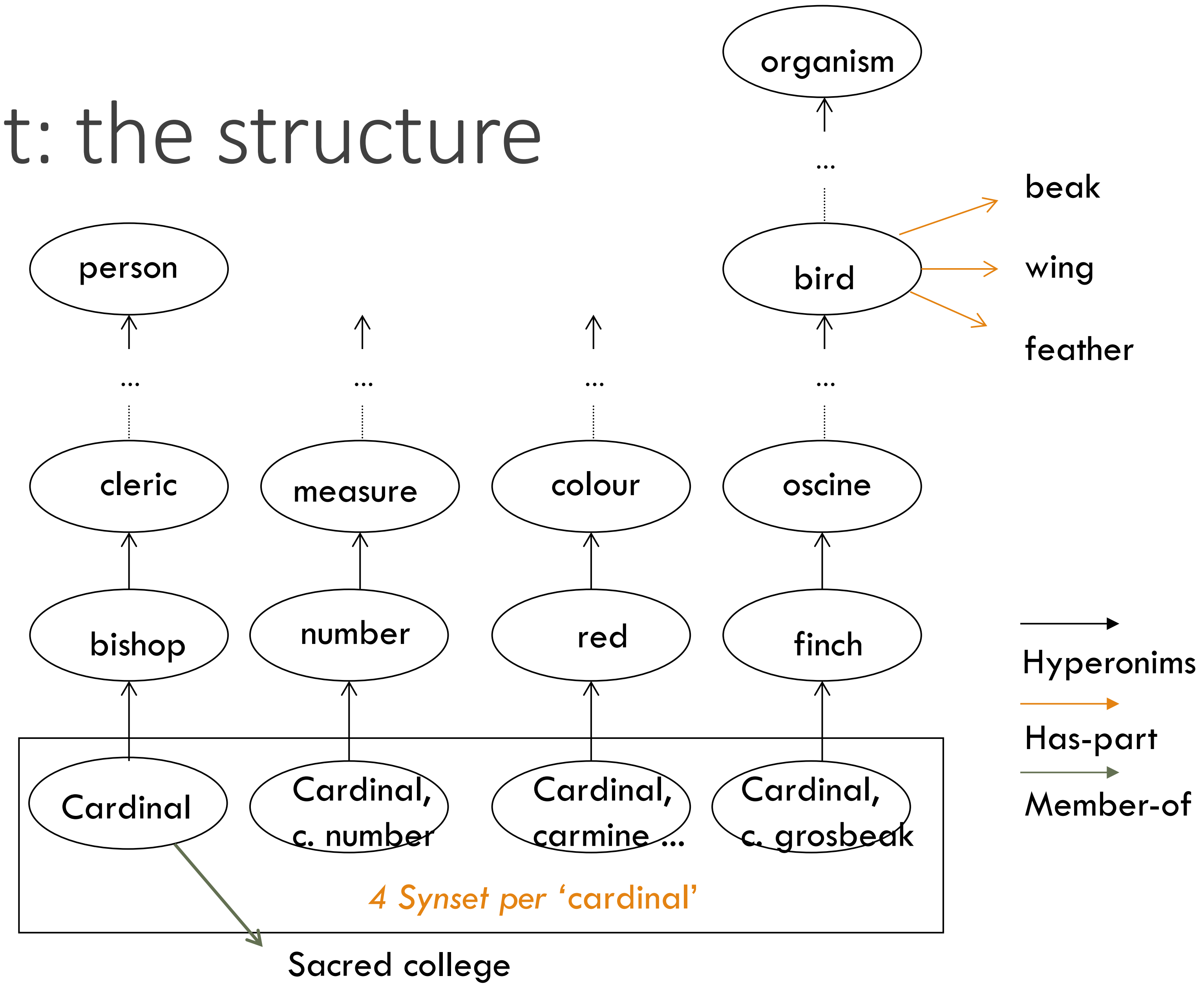
The Url of the project is:

- http://wordnet.princeton.edu/

Uses of WordNet in computational linguistics:

- Query expansion with synonyms or hyperonyms
- Semantics distance among words
- Semantic category of a word (or supersense)

WordNet: the structure



Knowledge graphs

Large-scale knowledge bases have been built, including:

- Open domain: Dbpedia, WikiData, Freebase, YAGO.
- Proprietary: Microsoft's Satori, Google's Knowledge Graph, Google’s Knowledge vault

From 2012 Google uses the **knowledge graph** with its search engine

- Structured data coming from many sources, including the *CIA World Factbook*, Wikidata, Wikipedia, Freebase.
- More than 70 billion facts in 2016

Since 2014, Google’s **Knowledge Vault** contains facts derived automatically from the web with machine learning techniques. Facts have an associated confidence value.

Towards **Giant Global Graph** (**GGG**) of the Semantic web [Tim Berners Lee, 2001] …

Object oriented representations and frames

Object Oriented representations

It is very natural to think of knowledge not as a flat collection of sentences, but rather as *structured* and *organized* in terms of the **objects** the knowledge is about.

- Complex objects have attributes, parts constrained in various ways
- Objects might have a behavior that is better expressed as **procedures**

… very much as in Object Oriented Programming

Marvin Minsky in 1975 suggested the idea of using as structured representation of objects, called **frames**, to recognize and deal with new situations.

Frames

Knowledge is organized in complex mental structures called **frames**

["*A Framework for Representing Knowledge*", Minsky, 1974].

The essence of the theory:

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

A frame is a data-structure for representing a **stereotypical situation**:

- Examples: hotel bedroom, or going to a child's birthday party.

Frames as data structures - 1

There are two types of frames:

1. **individual frames**, used to represent single objects
2. **generic frames**, used to represent categories or classes of objects.

An individual frame is a collection of *slot-fillers* pairs.

```
(Frame-name
    <slot-name1 filler1>
    <slot-name2 filler2>
    …)
```

Fillers can be:

- values, usually **default values**
- constraints on values
- the names of other individual frames
- the special slot INSTANCE-OF

Example 1

```
(tripLeg123
    <:INSTANCE-OF TripLeg>
    <:Destination toronto> . . . )

(toronto
    <:INSTANCE-OF CanadianCity>
    <:Province ontario>
    <:Population 4.5M> . . . )
```

Frames as data structures - 2

Generic frames are similar. Fillers can be:

- the special slot IS-A
- procedures
  1. IF-ADDED: activated when the slot receives a value
  2. IF-NEEDED: activated when the value is requested

These procedures are called **procedural attachments** or **demons**

The :INSTANCE-OF and :IS-A slots organize frames in frame systems.

They have the special role of activating inheritance of properties and procedures.

In frames, all values are understood as **default values**, which can be overridden.

```
(CanadianCity
    <:IS-A City>
    <:Province CanadianProvince>
    <:Country canada> )

(Lecture
   <:DayOfWeek WeekDay>
   <:Date  [IF-ADDED ComputeDayOfWeek ]>
   … )

(Table
   <:Clearance  [IF-NEEDED
   ComputeClearanceFromLegs ]> … )
```

Reasoning with frames

Attached procedures provide a flexible, organized framework for computation. Reasoning has a very procedural flavour.

A basic reasoning loop in a frame system has three steps:

1. **Recognition**: a new object or situation is recognized as instance of a generic frame;
2. **Inheritance**: any slot fillers that are not provided explicitly but can be inherited by the new frame instance are inherited;
3. **Demons**: for each slot with a filler, any inherited IF-ADDED procedure is run, possibly causing new slots to be filled, or new frames to be instantiated, until the system stabilizes; then the cycle repeats.

When the filler of a slot is requested:

1. if there is a value stored in the slot, the value is returned;
2. otherwise, any inherited IF-NEEDED procedure is run to compute the filler for the slot; this may cause other slots to be filled, or new frames to be instantiated.

Reasoning with frames: example

Chapter 8.3 of the book by Brachman & Levesque presents an example of **using frames to plan a trip** (made of travel steps).

I leave this as an exercise or presentation.

Frames and OOP

Frame-based representation languages and OOP systems were developed at the same time.

They look similar and certainly one could implement a frame system with OOP.

Important difference is that frame systems tend to work in a cycle:

- Instantiate a frame and declare some slot fillers
- inherit values from more general frames
- trigger appropriate forward-chaining procedures
- when the system is quiescent, stop and wait for the next input

The designer can control the amount of “forward” reasoning that should be done (in a *data-directed* fashion) or “backward” (in a *goal-directed* fashion).

Applications of frames

- Story understanding
- Scene recognition in vision
- Tools for building expert systems (possibly together with rules)
  Example: KEE (Fikes-Kehler, 85)
- The idea of frames was also used in building "FrameNet": a large NL resource based on the theory of "frame-based" semantics.

FrameNet [Lowe, Baker, Fillmore]

FrameNet is a resource consisting of collections of NL sentences syntactically and semantically annotated, organized in frames.

Frame semantics: the meaning of words emerges from the role they have in the conceptual structure of sentences.

Knowledge is structures in 16 general domains: time, space, communications, cognition, health …

6000 Lexical elements; 130.000 annotated sentences.

http://www.icsi.berkeley.edu/~framenet/

FrameNet: an example

FRAME: communication

FRAME DESCRIPTION: A person (COMMUNICATOR) produces some linguistic object (MESSAGE) while addressing some other person (ADDRESSEE) on some topic (TOPIC)

FE: COMMUNICATOR …

FE: MESSAGE …

FE: ADDRESSEE …

FE: TOPIC ..

[Pat] communicated [the message] [to me].

[Management] should develop and communicate [to all employees] [a vision of where the organization is going].

Videotapes of school activities are useful means of communicating [about work undertaken at school].

Conclusions

- ✓ Knowledge representation is **not only "logics"**. Logic is modular and well defined but has no structure.
- ✓ **Structure** can be exploited to reason more efficiently.
- ✓ Network based representations are very "natural" but have been used too informally: the definition of **legal conclusions** cannot be left to programs.
- ✓ **Procedural knowledge**, as used in frames, in addition to declarative knowledge, is still a viable alternative for reasoning systems.

Your turn

Proposals for assertional networks:

- ✓ Sowa’s conceptual graphs
- ✓ Tesnière dependency graphs and successors.
- ✓ Shapiro’s SNePS (Semantic Network Processing System)
- ✓ …

Frames based reasoning:

- ✓ Chapter 8.3 of the book by Brachman & Levesque: example of **using frames to plan a trip**
- ✓ FrameNet project
- ✓ KEE (Fikes-Kehler, 85)

References

[AIMA] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach* (3rd edition). Pearson Education 2010 (Ch. 12).

[KR&R] Ronald Brachman and Hector Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 2004 (Ch. 8-10)