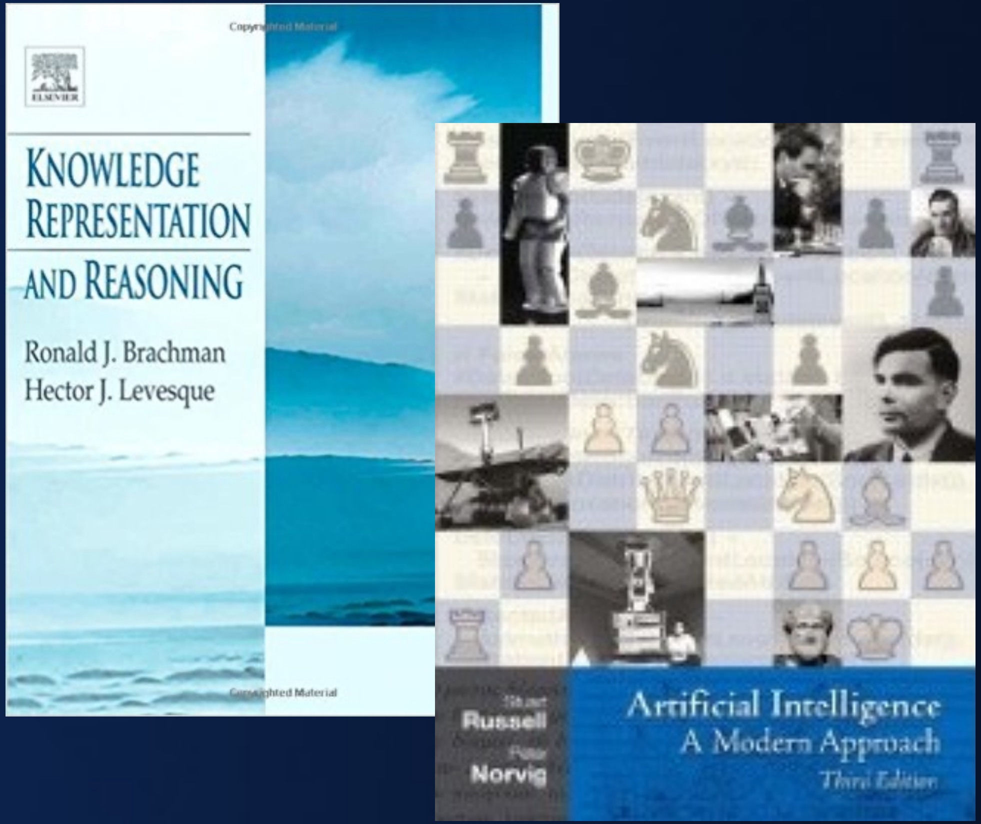


AI Fundamentals: Knowledge Representation and Reasoning

Maria Simi

Graph representations and structured representations

LESSON 5: SEMANTIC NETWORKS, FRAMES

Psychological-linguistic approach to KR&R

- The **logical approach**: suitable to model rational reasoning
  - ✓ Extensions to model commonsense reasoning
    - - *Frame* problem, nonmonotonic reasoning, propositional attitudes …
  - ✓ FOL contractions to control computational complexity of reasoning
    - - Description logics, logic programming …
- 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, such as cognitive psychology and linguistics

Associationist theories of KR

In logic systems, symbolic expressions are modular and are syntactically transformed without paying attention to the symbols used or to their “meaning”.

The symbol themselves are arbitrary.

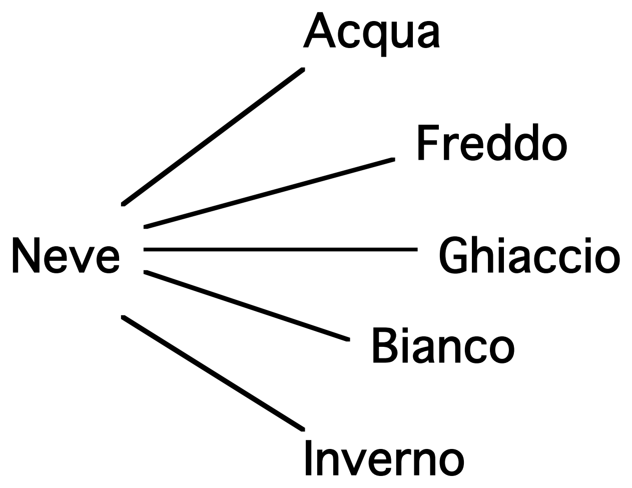
$$\forall x \ \ \text{Strawberry}(x) \Rightarrow \text{Red}(x)$$

It is all a matter of writing axioms to restrict interpretations

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.

Connections are shaped by experience, e.g. reading texts.



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 (e.g. *word embeddings* used in computational linguistics)
- Connectionist models for learning

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,

Nodes come in two flavors:

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

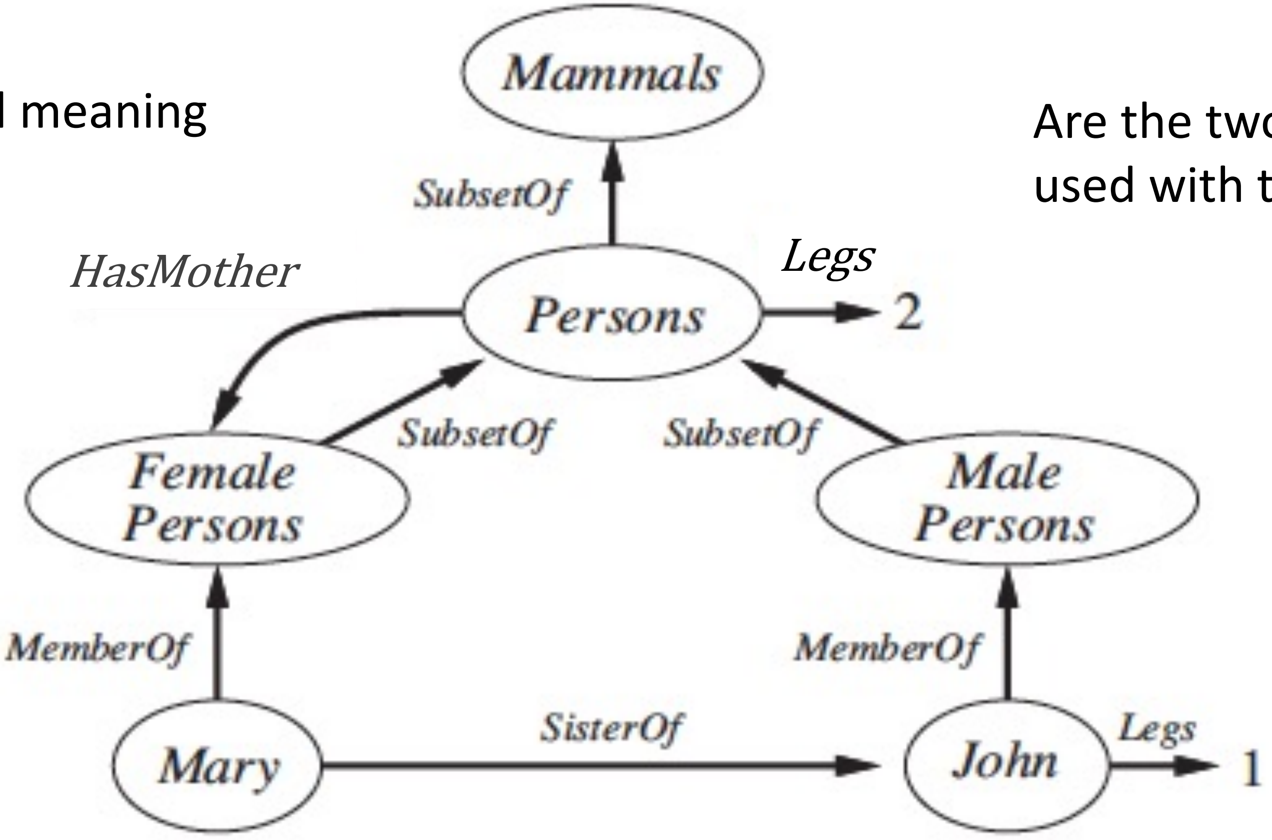
Two special kind of links are always present, with different names:

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*?

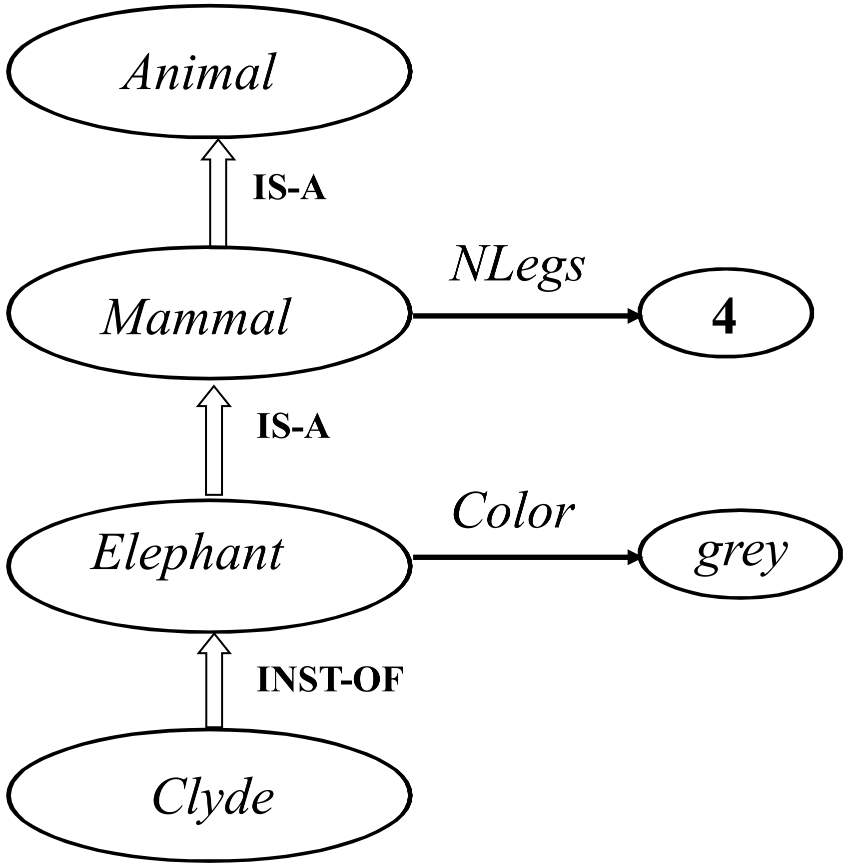
Inheritance in hierarchical networks

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 in OOP.

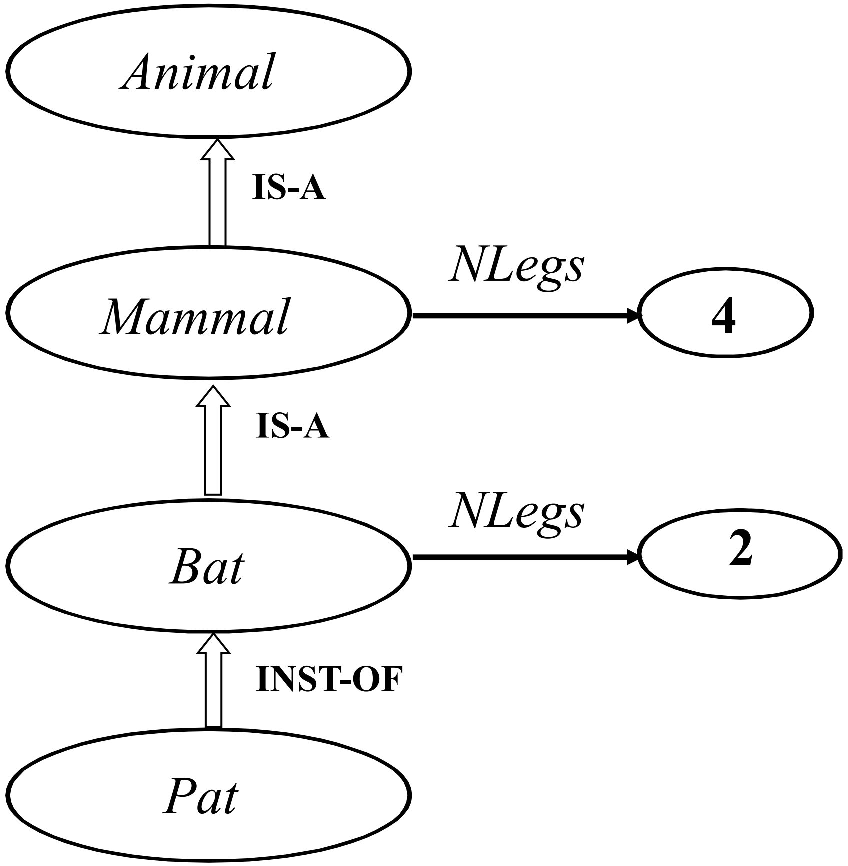


Inheritance with exceptions

The presence of exceptions does not create any problem.

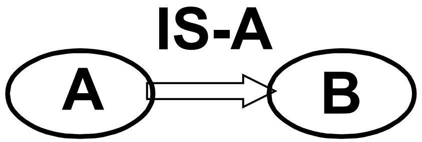
How many legs has Pat?

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



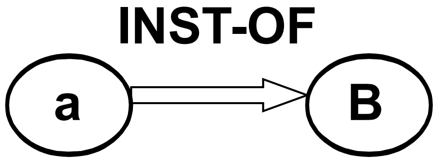
Logic account of semantic networks

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



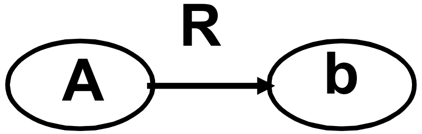
$\forall x\, A(x) \Rightarrow B(x)$

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



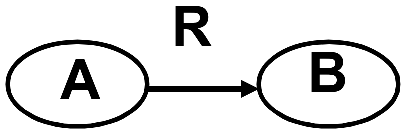
$B(a)$

a belongs to class B



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

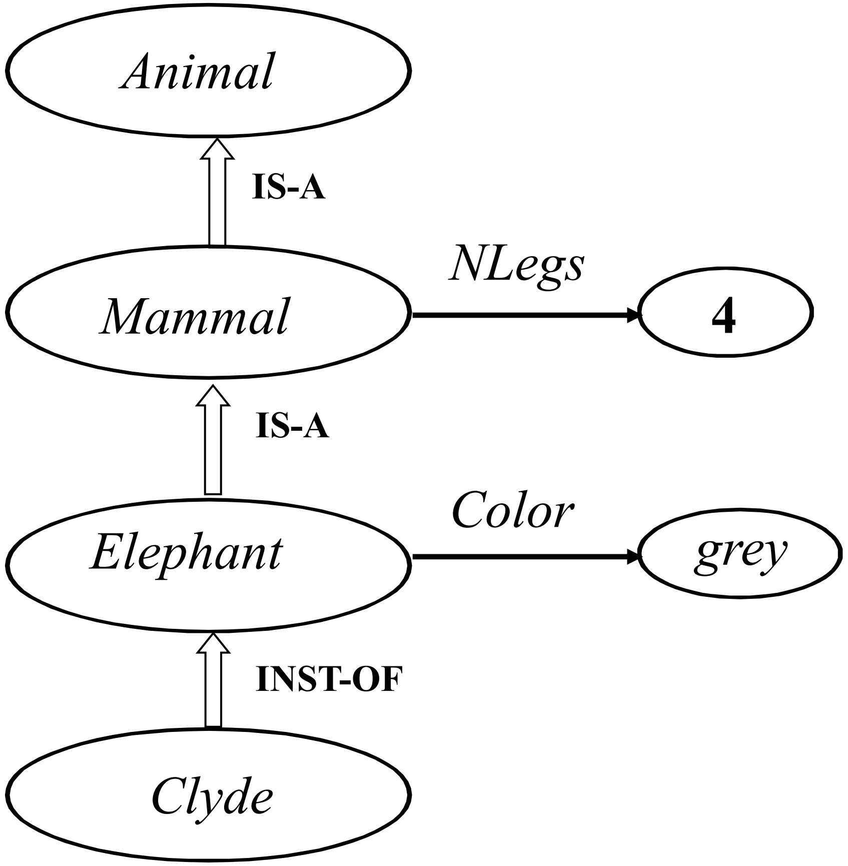
All members of class A are in relation R with b



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

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

Translation example



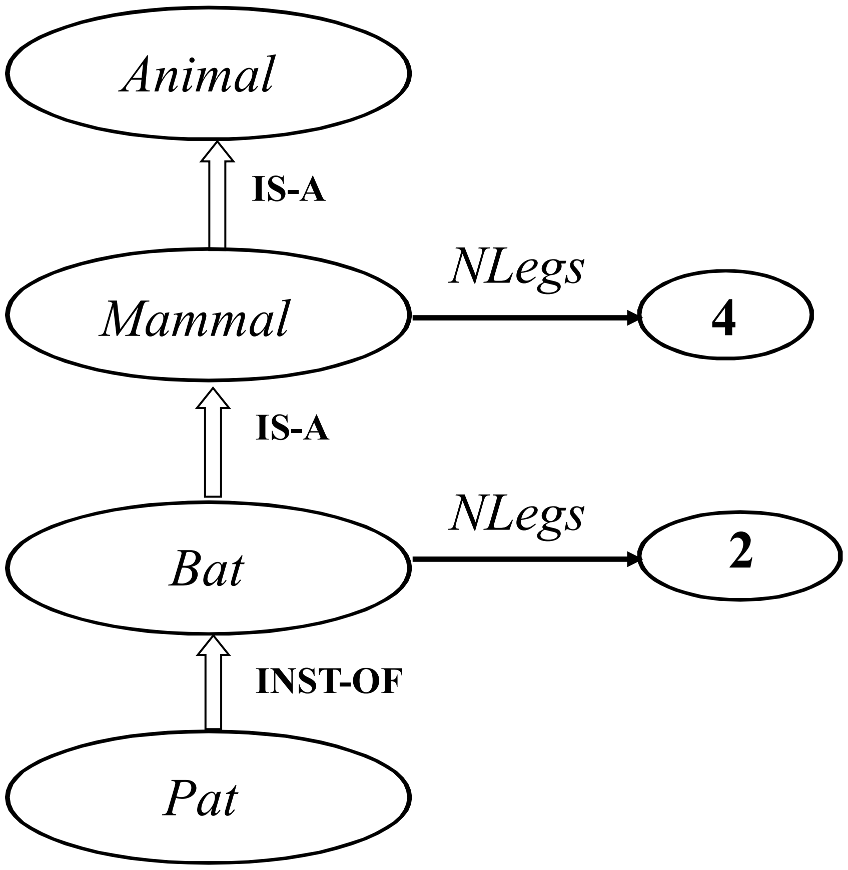
$\forall x\ Mammal(x) \Rightarrow Animal(x)$
$\forall x\ Mammal(x) \Rightarrow NLegs(x, 4)$
$\forall x\ Elephant(x) \Rightarrow Mammal(x)$
$\forall x\ Elephant(x) \Rightarrow Color(x, grey)$
$Elephant(Clyde)$

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 is a problem



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

It is possible to deduce:

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

and this may lead to a contradiction.

Rewriting the rule as:

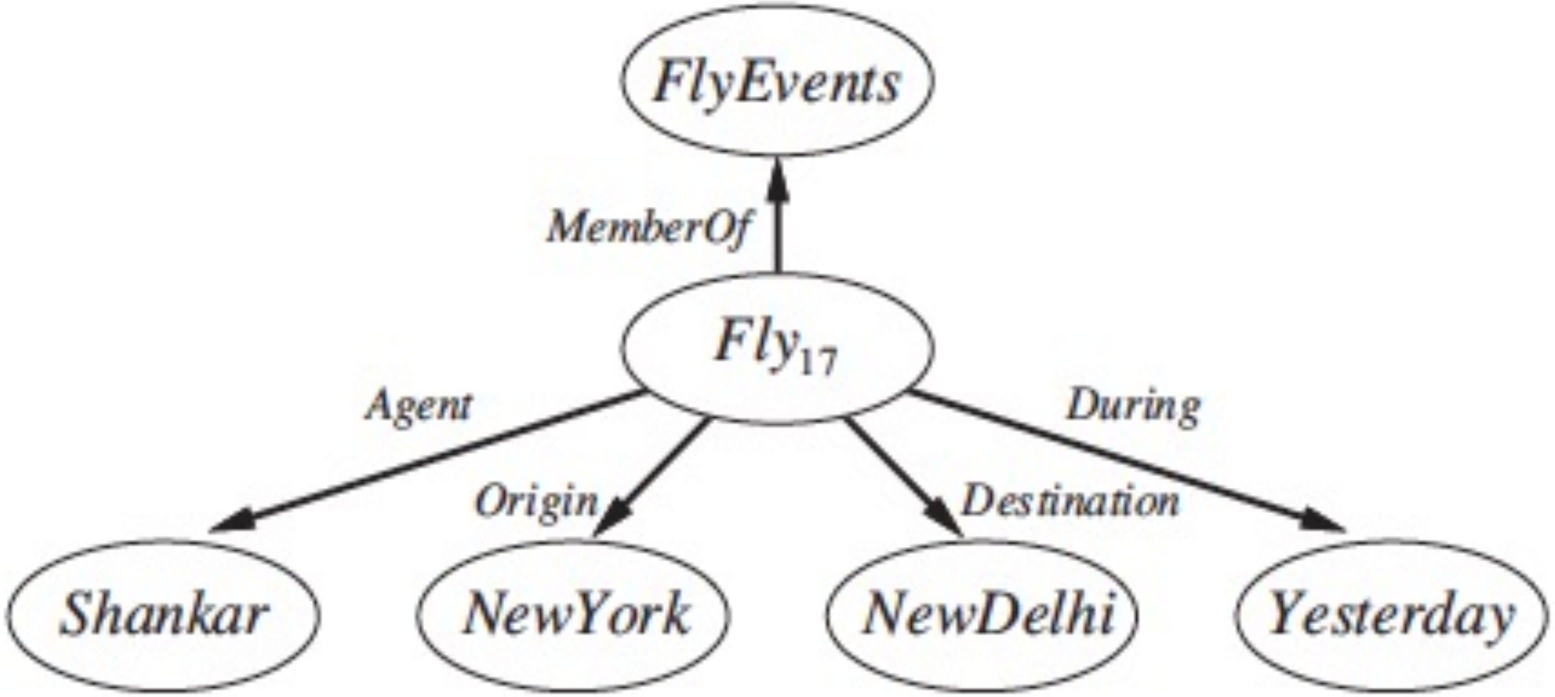
$\forall x\ Mammal(x) \land x \neq Pat \Rightarrow NLegs(x, 4)$

Problem: logics are **flat**.

Defaults require nonmonotonic reasoning

Only binary relations?

Case structure representation of «Shankar is flying from New York to New Deli yesterday”



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, $\vee$, $\Rightarrow$ ... 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" (existential graphs).
- 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.

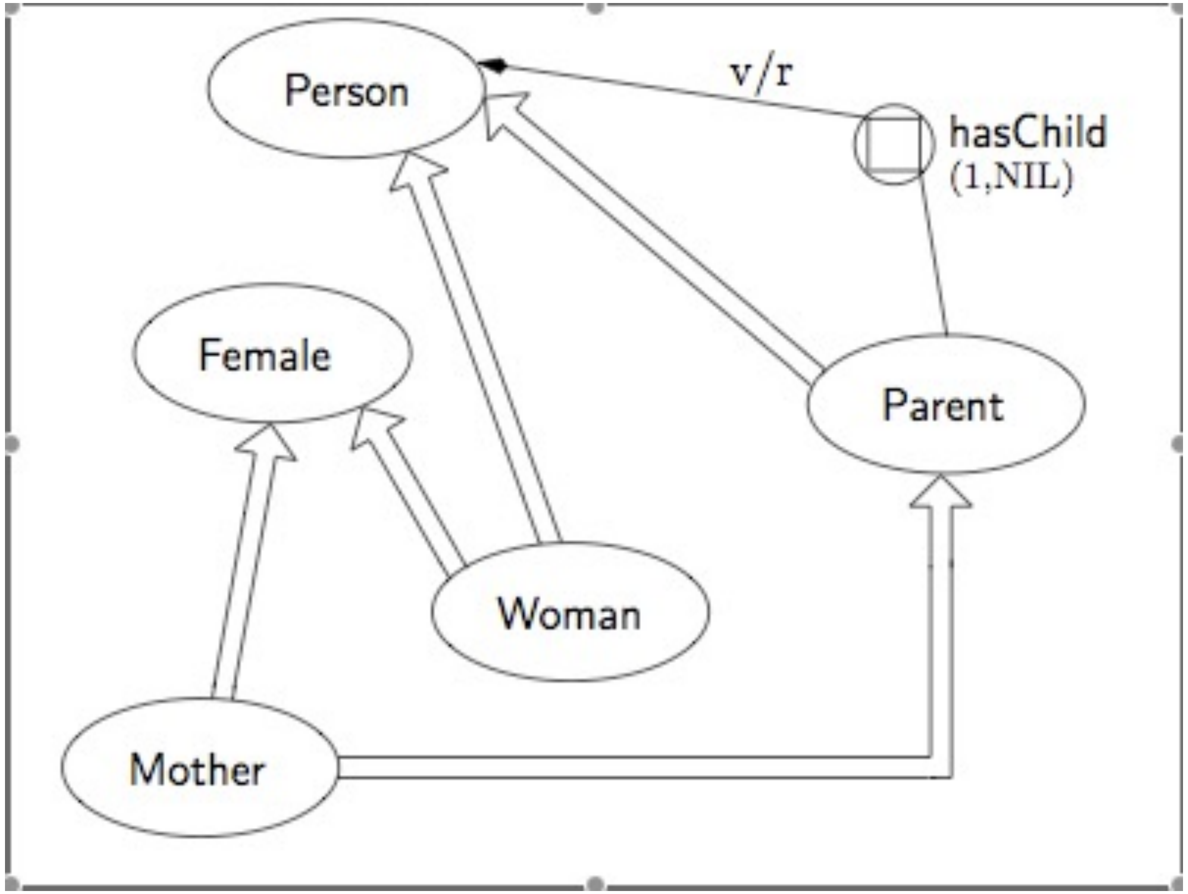
An example from KL-One

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.



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 a structured representation of objects, called **frames**, to recognize and deal with new situations.

Note: this has nothing to do with the *frame problem*.

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 object or 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*1 *filler*1>
<*slot*2 *filler*2>
...)

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 procedural flavor.

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.

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).

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 **correct 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.
- ✓ Next, we will see how these ideas have been formalized in description logics.

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)