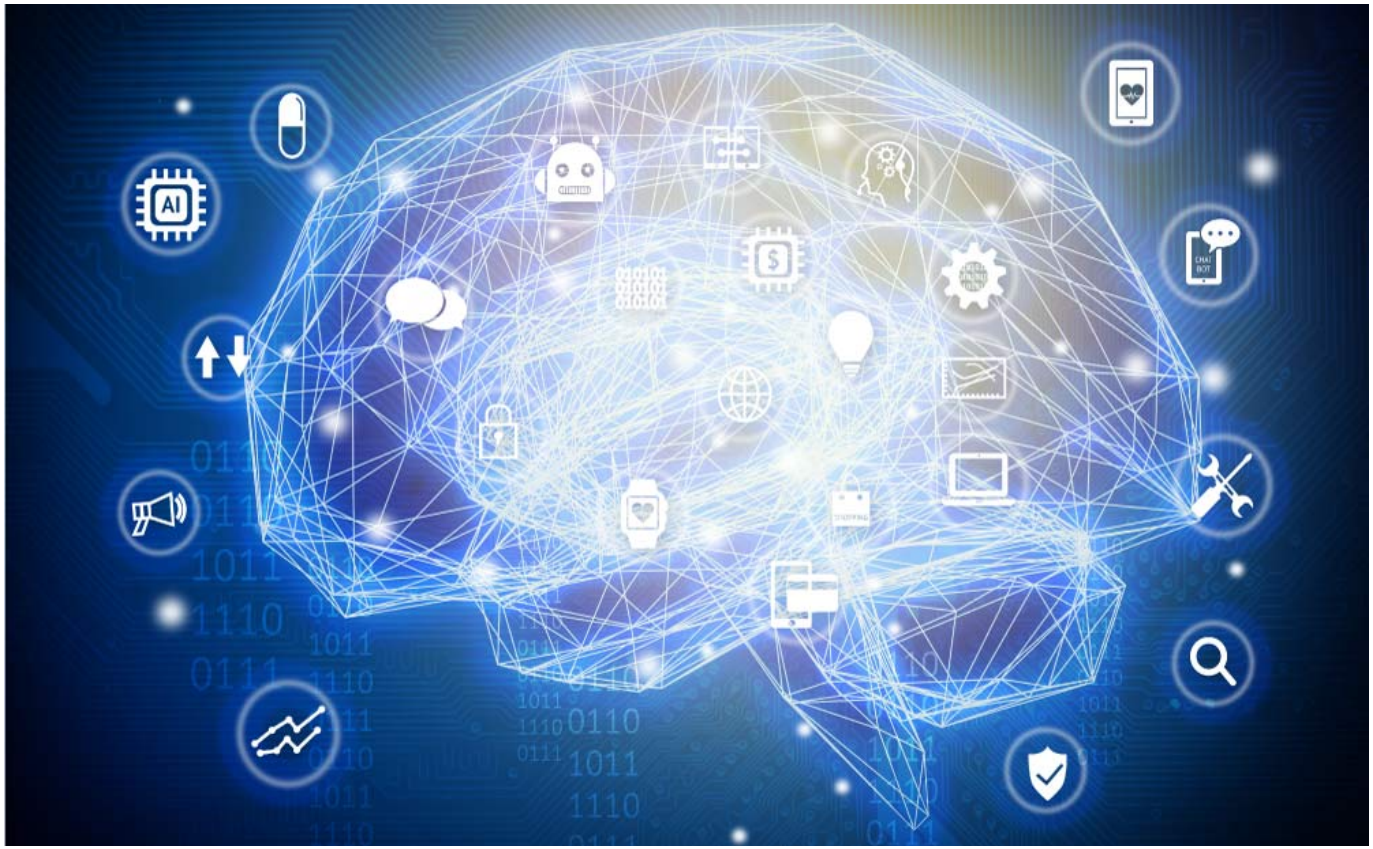


Reasoning Systems for Categories

Reasoning Systems for Categories

- Categories are the primary building blocks of large-scale knowledge representation schemes.

- This topic describes systems specially designed for organizing and reasoning with categories.

- There are two types of reasoning systems:
    1. **Semantic networks**
    2. **Description logics**

Reasoning systems

Semantic networks

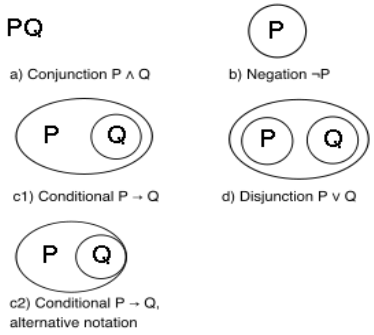
- Visualize knowledge-base in patterns of interconnected nodes and arcs
- Efficient algorithms for inferring of object on the basis of its category membership

Description logics

- Formal language for constructing and combining category definitions
- Efficient algorithms to decide subset and superset relationships between categories.

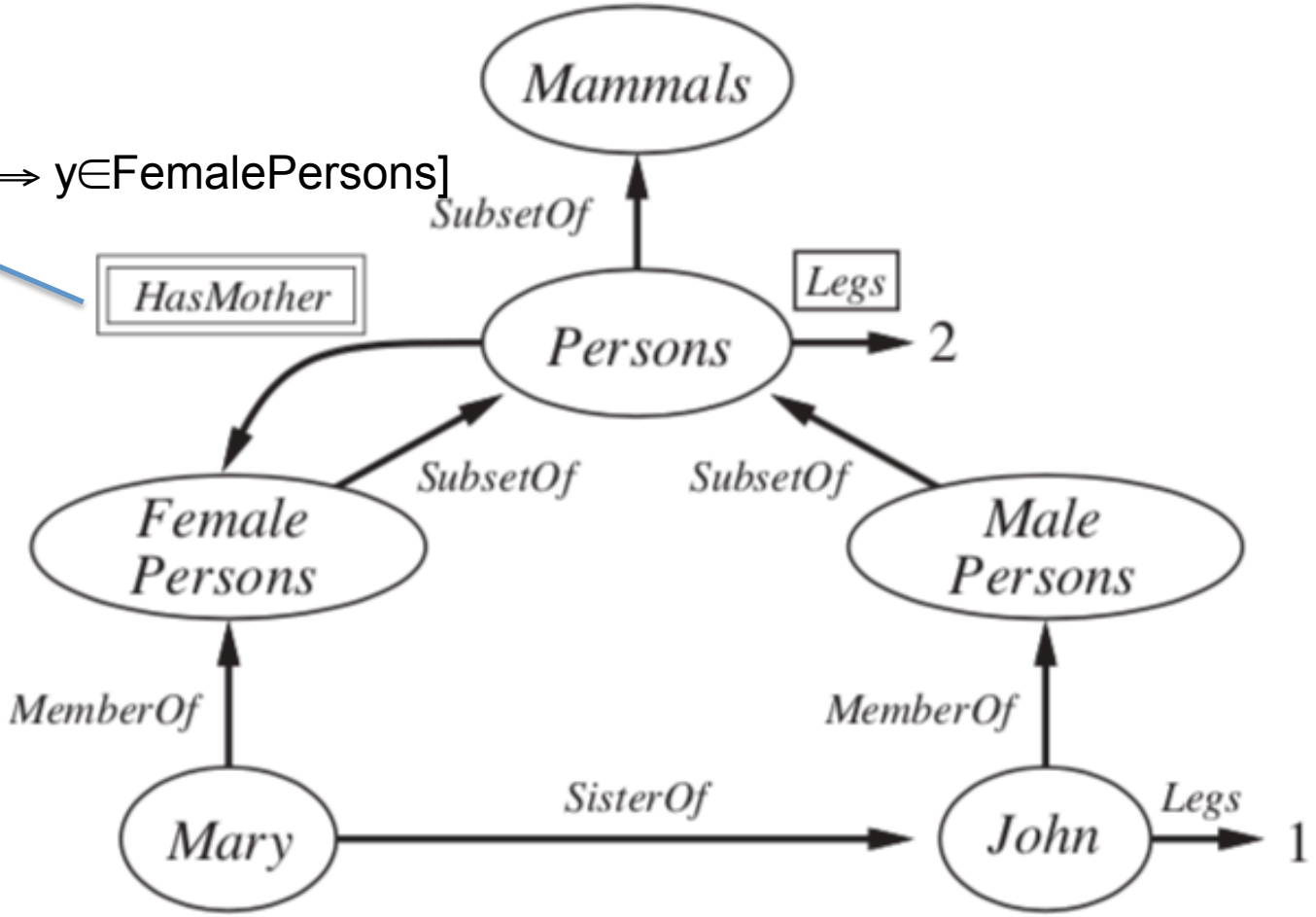
Semantic Networks

- In 1909, Charles S. Peirce proposed a graphical notation of nodes and edges called **existential graphs**
- A typical graphical notation displays object or category names in ovals or boxes, and connects them with labeled arcs/links
- For Example:
  - MemberOf link between Mary and FemalePersons, corresponding to the logical assertion **Mary ∈ FemalePersons**
  - SisterOf link between Mary and John corresponds to the assertion **SisterOf (Mary, John)**
  - connect categories using SubsetOf links,



Semantic Network Example

$\forall x\ x \in Persons \Rightarrow$
$[\forall y\ HasMother(x,y) \Rightarrow y \in FemalePersons]$



A semantic network with four objects (John, Mary, 1, and 2) and four categories. Relations are denoted by labeled links.

Semantic Networks

- Allows for inheritance reasoning
  - Female persons inherit all properties from person
  - Mary inherits the property of having two legs
- The simplicity and efficiency of this inference mechanism compared with logical theorem has been one of the main attractions of semantic networks.
- Multiple Inheritance becomes complicated because two or more conflicting values for answering the query
- For this reason, multiple inheritance is banned in some **object-oriented programming** (OOP) languages, such as Java

Semantic Networks

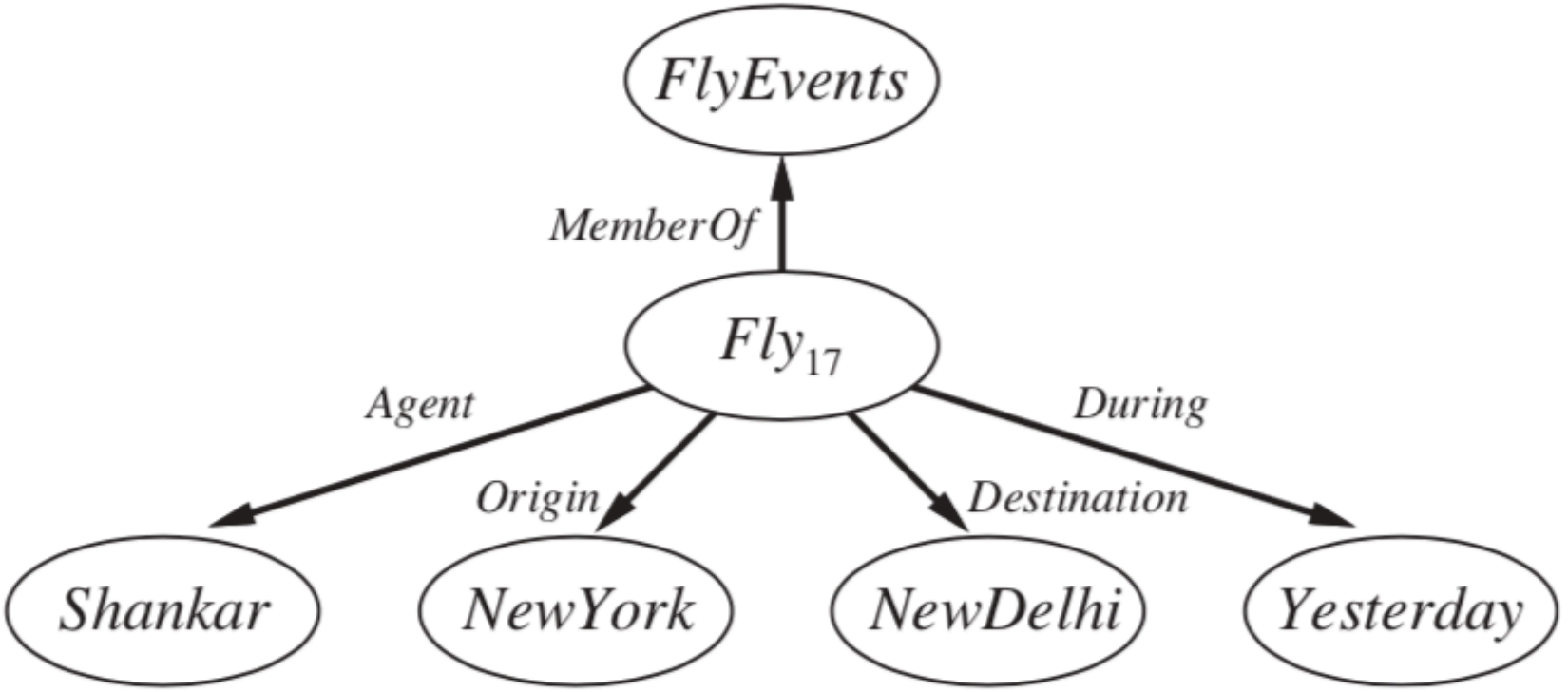
- Another form of inference is the use of inverse links
- Example: HasSister is the inverse of SisterOf

- Drawback of semantic network is that the links between bubbles represent only *binary* relations.

- For example, the sentence Fly(Shankar, NewYork, NewDelhi, Yesterday) cannot be asserted directly in a semantic network.

- *But can* obtain the effect of n-ary assertions by reifying the proposition

Semantic Networks



Semantic network showing representation of logical assertion fly()

Semantic Networks

- Ability to **override** the **default** values
- Example:
  - John has 1 leg despite the fact that all persons have 2 legs
  - This would be contradiction in a strictly logical KB.

Description logics

- They are logical notations that are designed to describe definitions and properties about categories
- It is to formalize the semantic network
- 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

Description logics

- The CLASSIC language is a typical description logic
- Any CLASSIC can be written in FOL
- For example, to say that bachelors are unmarried adult males we would write
    - **Bachelor = And (Unmarried , Adult , Male)**
- The equivalent in first-order logic would be
    - **$Bachelor(x) \Leftrightarrow Unmarried(x) \wedge Adult(x) \wedge Male(x)$**

The syntax of descriptions in a subset of the CLASSIC language.

$$
\begin{aligned}
\mathit{Concept} &\rightarrow \mathbf{Thing} \mid \mathit{ConceptName} \\
&\mid \mathbf{And}(\mathit{Concept}, \ldots) \\
&\mid \mathbf{All}(\mathit{RoleName}, \mathit{Concept}) \\
&\mid \mathbf{AtLeast}(\mathit{Integer}, \mathit{RoleName}) \\
&\mid \mathbf{AtMost}(\mathit{Integer}, \mathit{RoleName}) \\
&\mid \mathbf{Fills}(\mathit{RoleName}, \mathit{IndividualName}, \ldots) \\
&\mid \mathbf{SameAs}(\mathit{Path}, \mathit{Path}) \\
&\mid \mathbf{OneOf}(\mathit{IndividualName}, \ldots) \\
\mathit{Path} &\rightarrow [\mathit{RoleName}, \ldots]
\end{aligned}
$$