UNIVERSITÀ DEGLI STUDI DI NAPOLI
PARTHENOPE

Artificial Intelligence

Knowledge Representation

LESSON 18

prof. Antonino Staiano

M.Sc. In ''Machine Learning e Big Data'' - University Parthenope of Naples

Introduction

- An agent with a knowledge base can make inferences enabling it to act appropriately
- Now, the question is what content to put into an agent's knowledge base, that is, how to represent facts about the world
- We can use FOL as a representation language for discussing the content and organization of knowledge
  - Even though other representation formalisms exist

Outline

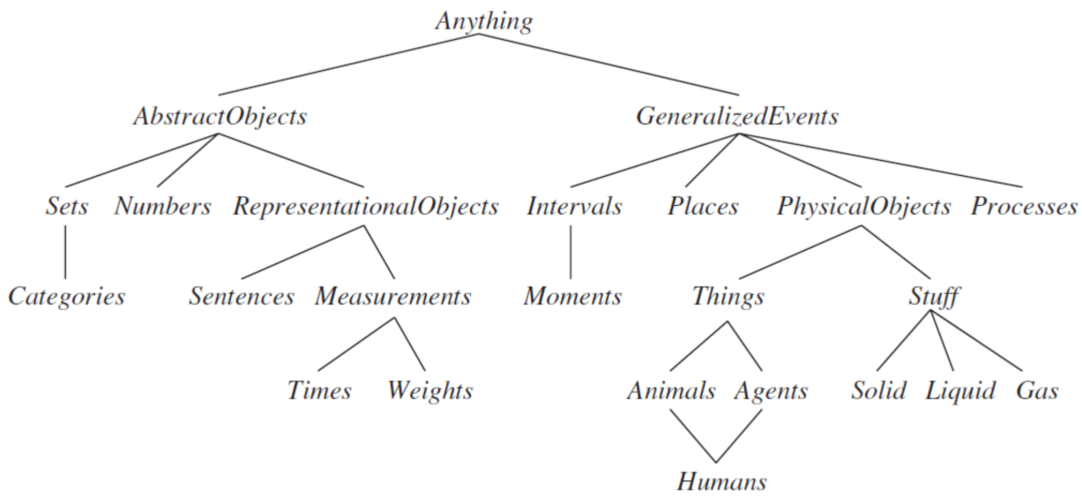
- Ontological Engineering
- Categories and Objects
- Events
- Mental Objects and Modal Logic
- Reasoning Systems for Categories

Ontologies

- An ontology formally represents knowledge that defines the concepts, relationships, and properties within a specific domain
- It provides a structured framework for organizing and sharing knowledge in a machine-readable format
- Ontologies play a crucial role in knowledge representation, semantic web technologies, and various fields of artificial intelligence

Ontological Engineering

- The representations can be created focusing on general concepts such as Events, Time, Physical objects, and belief
- Ontological Engineering
  - General and flexible representations for complex domains
- Upper ontology
  - The general framework of concepts
- Example
  - Ontology of the world



  - Each link indicates that the lower concept is a specialization of the upper one

Categories and Objects

- The organization of objects into categories is a vital part of knowledge representation
  - Indeed, much reasoning takes place at a level of categories
- Categories for FOL can be represented by predicates and objects
  1. BasketBall(s) is a predicate whereas
  2. a category can be reified as an object BasketBalls
     - Turning a proposition into an object
  - Member(s, BasketBalls) that is s is a member of the category BasketBalls
- We can use Subset(BasketBalls, Balls), that is, BasketBalls ⊂ Balls, to say that BasketBalls is a subcategory of Balls
- Categories organize knowledge through inheritance
  - If all instances of Food are edible, Fruit ⊂ Food and Apple ⊂ Fruit we infer that Apple is edible
- Subclass relations organize categories into a taxonomic hierarchy or taxonomy

Categories and Objects

- FOL makes it easy to state facts about categories
  - Relating objects to categories or quantifying their members
- Examples
  - An object is a member of a category
    - $BB9 \in Basketballs$
  - A category is a subclass of another category
    - $Basketballs \subset Balls$
  - All members of a category have some properties
    - $(x \in Basketballs) \Rightarrow Spherical(x)$
  - Members of a category can be recognized by some properties
    - $Orange(x) \wedge Round(x) \wedge Diameter(x)=9.5'' \wedge x \in Balls \Rightarrow x \in Basketballs$
  - A category as a whole has some properties
    - $Dogs \in DomesticatedSpecies$

Categories and Objects

- Categories can also be defined by providing necessary and sufficient conditions for membership
- Example
  - a bachelor is an unmarried adult male:
    - $x \in Bachelors \Leftrightarrow Unmarried(x) \wedge x \in Adults \wedge x \in Males$

Categories and Objects

- Physical composition: the idea that an object can be part of another is a familiar one
  - We use the general PartOf relation to say that one thing is part of another
    - Eg: Bucharest is part of Romania, PartOf(Bucharest, Romania)
- Objects can be grouped into PartOf hierarchies
  - PartOf(Bucharest,Romania)
  - PartOf(Romania,EasternEurope)
  - PartOf(EasternEurope,Europe)
  - PartOf(Europe,Earth)
- The PartOf relation is transitive and reflexive, that is,
  - $PartOf(x,y) \land PartOf(y, z) \Rightarrow PartOf(x, z)$
  - $PartOf(x,x)$

Categories and Objects

- In both scientific and common-sense theories of the world, objects have height, mass, cost, and so on
  - The values we assign to these properties are called measures
  - Ordinary quantitative measures are easy to represent
- We imagine that the universe contains abstract "measure objects", such as length, that is the length of a line segment (denote it by L1)
  - We represent length with a units function that takes a number as an argument
    - Eg: Length(L1) = Inches(1.5) = Centimeters(3.81)
    - The same length has different names in our language

Things and Stuff

- The real world can be viewed as consisting of primitive objects (e.g., atomic particles) and composite objects built from them as an aggregation
- By keeping the reasoning at the level of objects as large as apples and cars, we can handle the complexity inherent in treating many primitive objects individually
- However, a significant portion of reality appears to resist clear division into distinct objects, and this is referred to as stuff
  - As an example, if we have some butter there is no obvious number of “butter-objects” because any part of a butter-object is also a butter-object
- Instead, If we have a cat, we have just one cat and if we cut the cat, we do not get two cats
  - the cat is an example of a thing

Categories and Objects

- Natural language distinguishes clearly between stuff and things
  - we say a cat (count nouns, e.g., holes, theorems, …) and not a butter (mass nouns, e.g., water, energy, …)
- Some properties are intrinsic: they belong to the very substance of the object, rather than to the object as a whole
  - When you cut an instance of stuff in half, the two pieces retain the intrinsic properties, such as density, boiling point, flavor, color, ownership …
  - Substance
    - a category of objects that includes in its definition only intrinsic properties (mass noun)
- Other properties are extrinsic
  - Properties like weight, length, and shape that are not retained under subdivision
  - Count noun
    - A class that includes any extrinsic properties in its definition

Events

- Something that happen
  - Events (*things that happen*), fluents (*aspects of the world that change*), time points
  - It is possible to represent events and fluents with propositions, but it is easy in a world where events are discrete, instantaneous, happen one at a time, and have no variations …
- Event calculus
  - Approach to describe what's happening during an event or action and two actions happening at the same time
  - The objects of event calculus are events, fluents, and time points
    - Fluents: At(Shankar, Berkeley), that is, the fact that Shankar is in Berkeley
    - Events: Event E1 of Shankar flying from San Francisco to DC
      - $E1 \in Flyings \wedge Flyer(E1, Shankar) \wedge Origin(E1, SF) \wedge Destination(E1, DC)$
    - Flyings is the category of all flying events
- To assert that a fluent is actually true starting at some point in time t_1 and continuing to time t_2
  - predicate T, as in $T(At(Shankar,Berkeley),t_1,t_2)$
- to say that the event E_1 actually happened, starting at time t_1 and ending at time t_2
  - Similarly, we use the predicate $Happens(E_1,t_1,t_2)$

Events

- Time points and time intervals

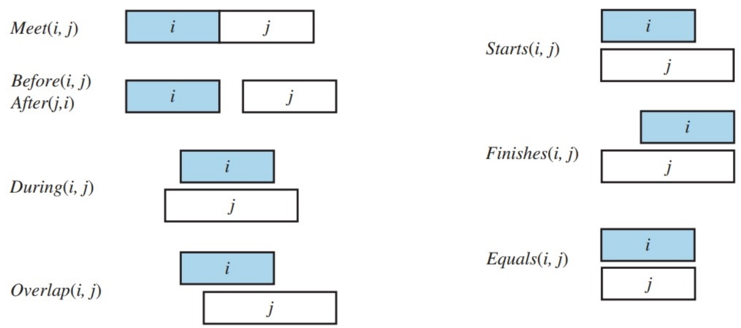


Figure 10.2 Predicates on time intervals.

$$
\begin{aligned}
Meet(i,j) &\Leftrightarrow End(i) = Begin(j)\\
Before(i,j) &\Leftrightarrow End(i) < Begin(j)\\
After(j,i) &\Leftrightarrow Before(i,j)\\
During(i,j) &\Leftrightarrow Begin(j) < Begin(i) < End(i) < End(j)\\
Overlap(i,j) &\Leftrightarrow Begin(i) < Begin(j) < End(i) < End(j)\\
Starts(i,j) &\Leftrightarrow Begin(i) = Begin(j)\\
Finishes(i,j) &\Leftrightarrow End(i) = End(j)\\
Equals(i,j) &\Leftrightarrow Begin(i) = Begin(j) \land End(i) = End(j)
\end{aligned}
$$

- To say that the reign of Elizabeth II immediately followed that of George VI, and the reign of Elvis overlapped with the 1950s, we can write the following:
  - Meets(ReignOf(GeorgeVI),ReignOf(ElizabethII))
  - Overlap(Fifties,ReignOf(Elvis))
  - Begin(Fifties) = Begin(AD1950)
  - End(Fifties) = End(AD1959)

Mental Objects

- Mental objects refer to abstract entities that exist within the realm of human cognition and mental representation
  - They are subjective constructs that represent various aspects of knowledge, beliefs, concepts, and ideas that individuals hold in their minds
  - Mental objects are knowledge in someone's head (or KB)
- Propositional attitudes refer to mental states or attitudes that individuals have toward propositions or statements
  - These attitudes represent an individual's beliefs, desires, intentions, opinions, and other mental states regarding the truth or falsehood of propositions
  - For instance, attitudes such as Believes, Knows, Wants, and Informs
- Example: Lois knows that Superman can fly:

  Knows(Lois, CanFly(Superman))

Modal Logic

- Sentences can sometimes be verbose and clumsy
  - Regular logic is concerned with a single modality, the modality of truth
- Modal logic addresses this, with special modal operators that take sentences (rather than terms) as arguments
  - Modal logic allows for reasoning about statements and propositions that are qualified by modalities, which reflect different modes of truth and possibility
  - It provides a framework to analyze and reason about concepts like necessity, possibility, impossibility, certainty, belief, and knowledge
- "A knows P" is represented with the notation $K_A P$, where K is the modal operator for knowledge
  - It takes two arguments, an agent (written as the subscript) and a sentence
- The syntax of modal logic is the same as first-order logic, except that sentences can also be formed with modal operators

Mental Objects and Modal Logic

- Having a modal operator for knowledge, one can write axioms for it
  - Agents are able to draw conclusions
    - If an agent knows P and knows that P implies Q, then the agent knows Q:

$$(K_A P \land K_A(P \Rightarrow Q)) \Rightarrow K_A Q$$

  - Logical agents can introspect on their own knowledge
    - If they know something, then they know that they know it:

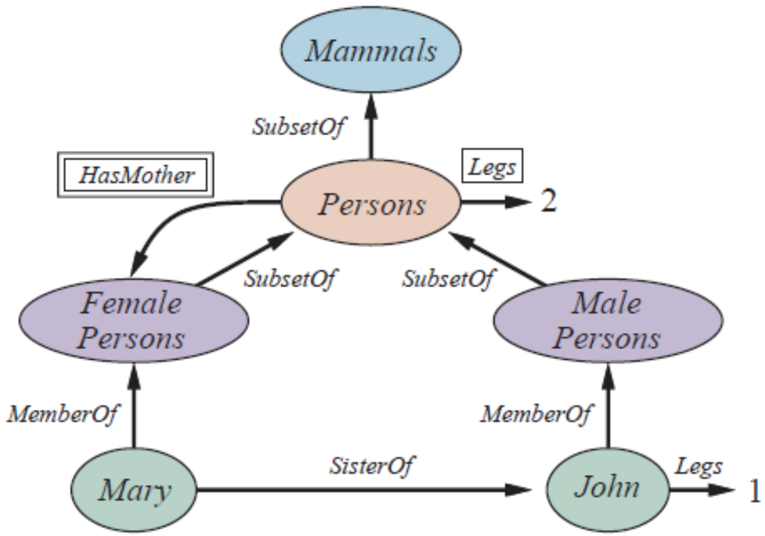
$$K_A P \Rightarrow K_A\,(K_A P)$$

Reasoning Systems for Categories

- Categories are the primary building blocks of large-scale knowledge representation schemes
- To organize and reason with categories there are two closely related families of systems
  - semantic networks
    - provide a graphical representation of a knowledge base and efficient algorithms for inferring the properties of an object based on its membership in a category
  - description logics
    - provide a formal language for constructing and combining category definitions and provide efficient algorithms for determining subset and superset relationships of categories

Reasoning Systems for Categories

- Semantic networks
  - Represent individual objects, categories of objects, and relations among objects
  - A typical graphical notation displays object or category names in ovals or boxes and connects them with labeled links
  - convenient to perform inheritance reasoning
    - Mary inherits the property of having two legs
      - to find out how many legs Mary has, the inheritance algorithm follows the MemberOf link from Mary to the category she belongs to
      - and then follows SubsetOf links up the hierarchy until it finds a category for which there is a boxed Legs link



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

$\forall x \;\; x \in Persons \Rightarrow Legs(x,2)$

Semantic Networks

- One drawback of semantic networks is that links between bubbles represent only binary relations
- Example
  - *Fly(Shankar, NewYork, NewDelhi, Yesterday)*
    - N-ary assertions by reifying the proposition itself as an event belonging to an event category

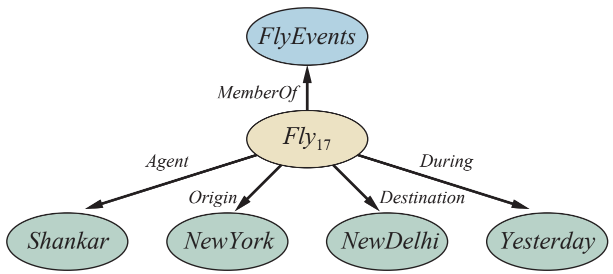


Figure 10.5 A fragment of a semantic network showing the representation of the logical assertion $Fly(Shankar, NewYork, NewDelhi, Yesterday)$.

Reasoning Systems for Categories

- Description logics
  - notations that are designed to make it easier to describe definitions and properties of categories
  - evolved from semantic networks in response to the need to formalize what the networks mean, while retaining the emphasis on taxonomic structure as an organizing principle
  - Principal inference tasks:
    - Subsumption: checking if one category is a subset of another by comparing their definitions
    - Classification: checking whether an object belongs to a category
- The CLASSIC language (Borgida et al., 1989) is a typical description logic
  - Eg: bachelors are unmarried adult males

    $$Bachelor = And(Unmarried, Adult, Male)$$

  - The equivalent in first-order logic would be:

    $$Bachelor(x) \Leftrightarrow Unmarried(x) \wedge Adult(x) \wedge Male(x)$$

Reasoning Systems for Categories

- The description logic has an algebra of operations on predicates, which we can't do in first-order logic
- Any description in CLASSIC can be translated into an equivalent first-order sentence, but some descriptions are more straightforward in CLASSIC
- Example
  - to describe the set of men with at least three sons who are all unemployed and married to doctors, and at most two daughters who are all professors in physics or math departments, we would use
    - And(Man,AtLeast(3,Son),AtMost(2,Daughter), All(Son,And(Unemployed,Married, All(Spouse,Doctor))), All(Daughter,And(Professor,Fills(Department,Physics,Math))))
- The language does not allow one to state that one concept or category is a subset of another

Concept	→	**Thing** | *ConceptName*
	|	**And**(*Concept*, …)
	|	**All**(*RoleName*, *Concept*)
	|	**AtLeast**(*Integer*, *RoleName*)
	|	**AtMost**(*Integer*, *RoleName*)
	|	**Fills**(*RoleName*, *IndividualName*, …)
	|	**SameAs**(*Path*, *Path*)
	|	**OneOf**(*IndividualName*, …)
Path	→	[*RoleName*, …]
ConceptName	→	*Adult* | *Female* | *Male* | …
RoleName	→	*Spouse* | *Daughter* | *Son* | …

Figure 10.6 The syntax of descriptions in a subset of the CLASSIC language.

Description Logic

- The most important aspect of description logics is their emphasis on the tractability of inference
  - A problem instance is solved by describing it and then asking if it is subsumed by one of several possible solution categories
- First-order logic systems often struggle to predict problem-solving times
  - requiring users to engineer problem representations to circumvent issue sets that cause the system to take weeks to solve
- The thrust for description logic is to ensure that subsumption testing can be solved in polynomial time to the size of the description