Ontological Engineering

Knowledge Representation

- Knowledge-representation is the field of AI dedicated to representing information about the world in a form that a computer system can utilize to solve complex tasks.

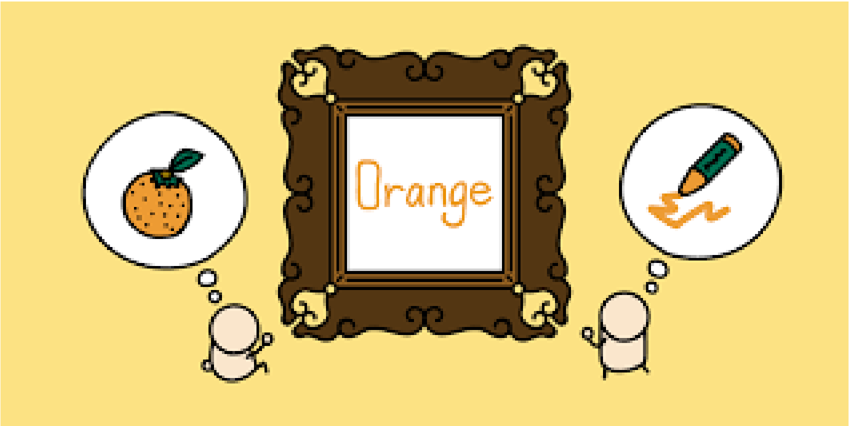
- This topic addresses what *content* to put into knowledge base

- How to represent facts about the world?

Ontological engineering

- Ontology engineering is a set of tasks related to the development of ontologies for a particular domain.

- Google definition: a set of concepts and categories in a subject area or domain that shows their properties and the relations between them.

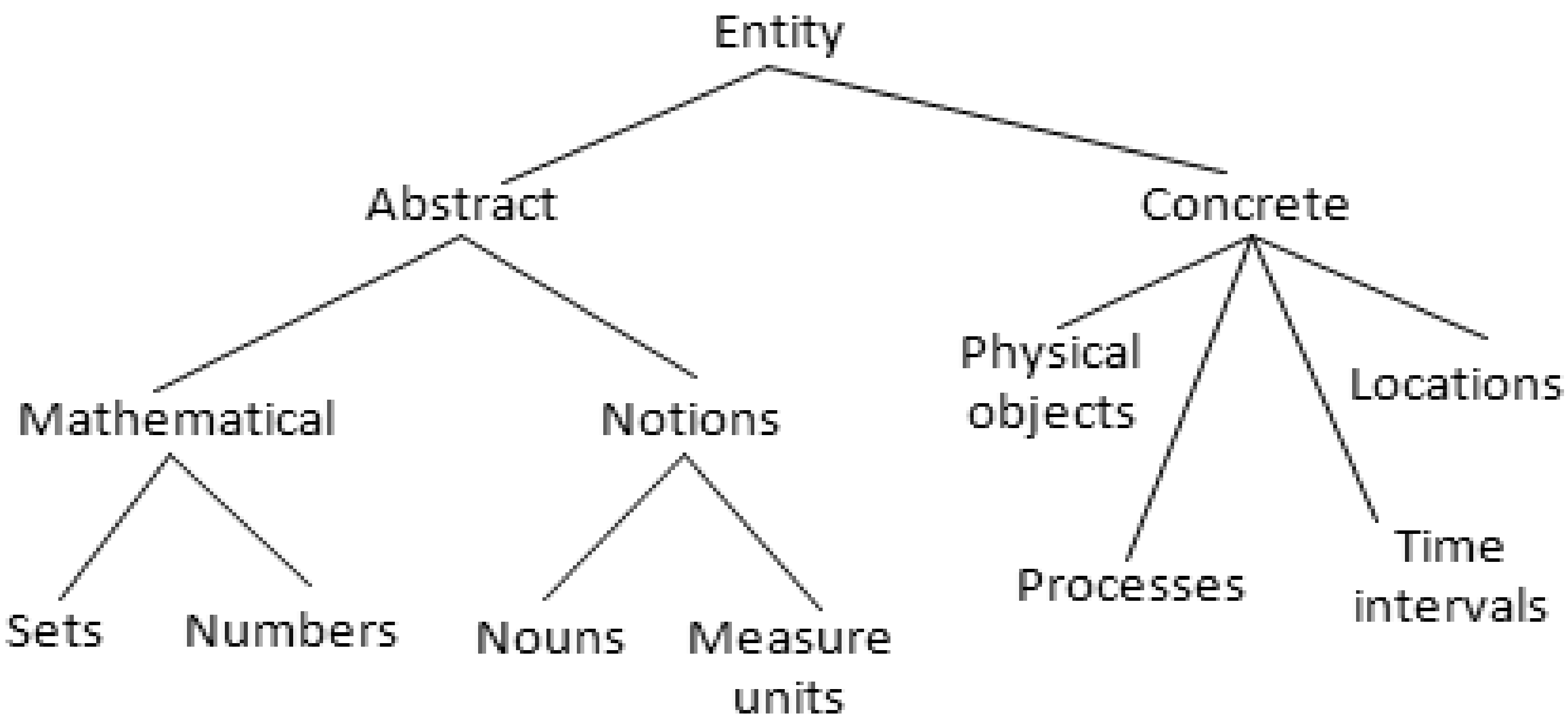


Ontological engineering

- How to create more general and flexible representations
  - Concepts like actions, time, physical objects and beliefs
  - Operates on a bigger scale than knowledge engineering
- Representing these abstract concepts is sometimes called **ontological engineering**.
- Define general framework of concepts (because representing everything is challenging) called as **upper ontology** with general concepts at the top and more specific concepts below the hierarchy
- Limitations of logic representation
  - Red, green and yellow tomatoes: exceptions and uncertainty

Ontological engineering

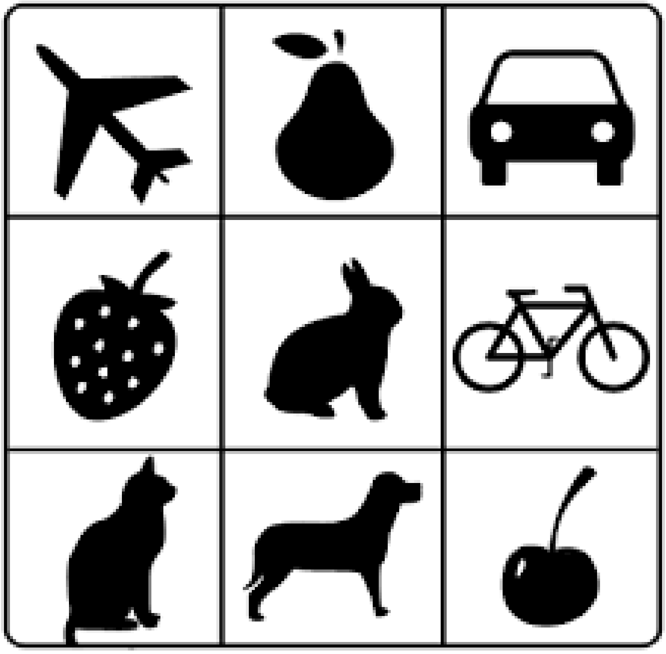
- Defining terms in the domain and relations among them
  - Defining concepts in the domain(classes)
  - Arranging the concepts in a hierarchy(subclass-superclass hierarchy)
  - Defining which attributes and properties classes can have and constraints on their values
  - Defining individuals and filling in property values



The upper ontology of the world. Each link indicates that the lower concept is a specialization of the upper one.

General-purpose ontology

- 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

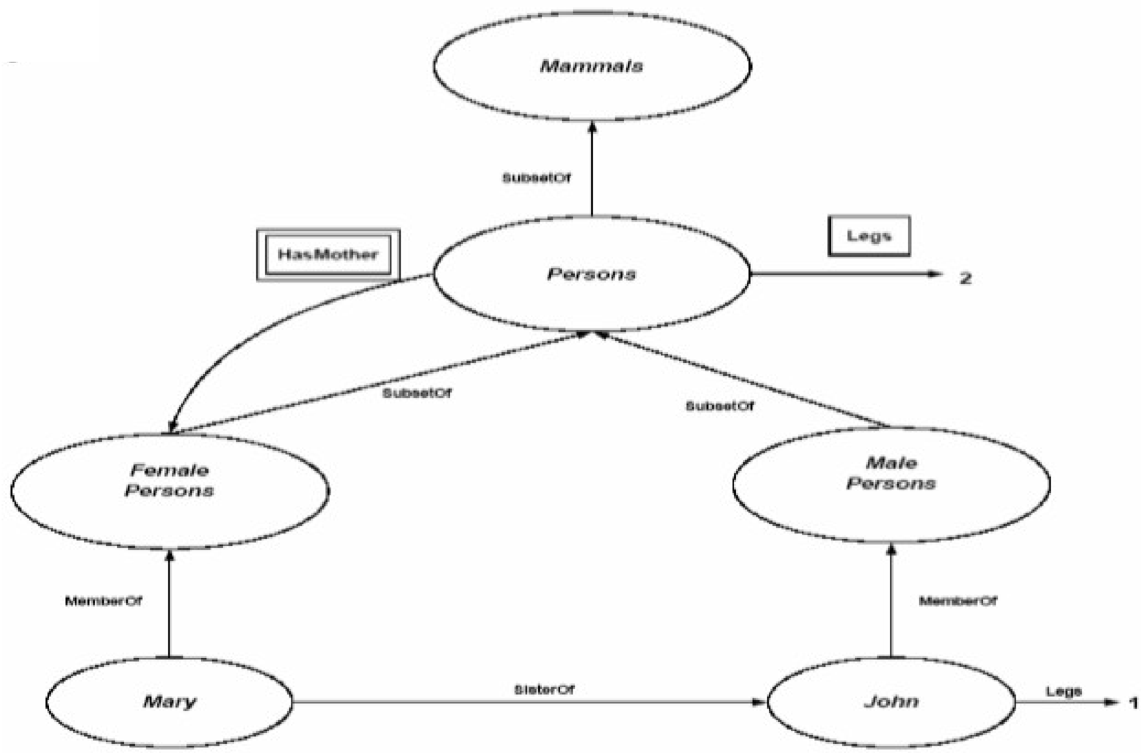
Categories and objects

- KR requires the organization of objects into categories. Although
  - 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
  1. **Predicates**: ***apple(x)***
  2. Reification of categories into **objects**: ***apples***
- Category = set of its members
  - Example: Member(x, apples), $x \in$ apples,
  - Subset(apples, fruits), apples $\subset$ fruits

Reify: make (something abstract) more concrete or real.

Category Organization

- Categories serve to organize and simplify the knowledge base through **inheritance**.
- 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.
  - Individual apples **inherit** the property of edibility from food
- Defines a taxonomy
  - Subclass relations organize categories



FOL and Categories

- First-order logic makes it easy to state facts about categories, either by relating objects to categories or by quantifying over their members.

Example:

- An object is a member of a category
  - MemberOf(BB12,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 are mutually exclusive
  - Disjoint({Animals, Vegetables})
- A decomposition of a class into categories is called **exhaustive** if each object of the class must belong to at least one category
  - living = {animal, vegetable, fungi, bacteria}
- A **partition** is an exhaustive decomposition of a class into disjoint subsets.
  - student = {undergraduate, graduate}

Natural kinds

- Many categories have no clear-cut definitions (e.g., chair, bush, book).
- Tomatoes: sometimes green, orange, red, yellow. Mostly spherical, smaller, larger etc
- One solution: Separate what is true of all instances of a category from what is true only of typical instances.
- subclass using category **Typical(Tomatoes)**
  - **$Typical(c) \subseteq c$**
- **$\forall x, x \in Typical(Tomatoes) \Rightarrow Red(x) \land Spherical(x)$**
- This helps us to write down useful facts about categories without providing exact definitions.

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)**

Logical Minimization: Defining an object as the smallest one satisfying certain conditions.

Measurements

- Objects have height, mass, cost,....
  - Values that we assign to these properties are called measures
- Combine Unit functions with a number to measure line segment object L1
  - Length(L1) = 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 is not numerical values, but measures can be orderable.
  - An apple can have deliciousness .9 or .1

Objects: Things and stuff

- Stuff: Defy any obvious **individuation** after division into distinct objects (mass nouns)
- Things: Individuation(count nouns)

- Example: Butter(stuff) and Cow(things)

$$b \in Butter \wedge PartOf(p, b) \Rightarrow p \in Butter$$

References

- Textbook
- https://www.cs.iusb.edu/~danav/teach/c463/10_knowledge.html