



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

---

Intelligenza Artificiale

# Knowledge representation and reasoning

## Ontologies and knowledge base representation

Ivan Heibi

Dipartimento di Filologia Classica e Italianistica (FICLIT)

[Ivan.heibi2@unibo.it](mailto:Ivan.heibi2@unibo.it)

---

# Ontologies

*What content to put into an agent's knowledge base and how to represent facts about the world?*

Complex domains (e.g., shopping on the Internet) require more general and flexible representations.

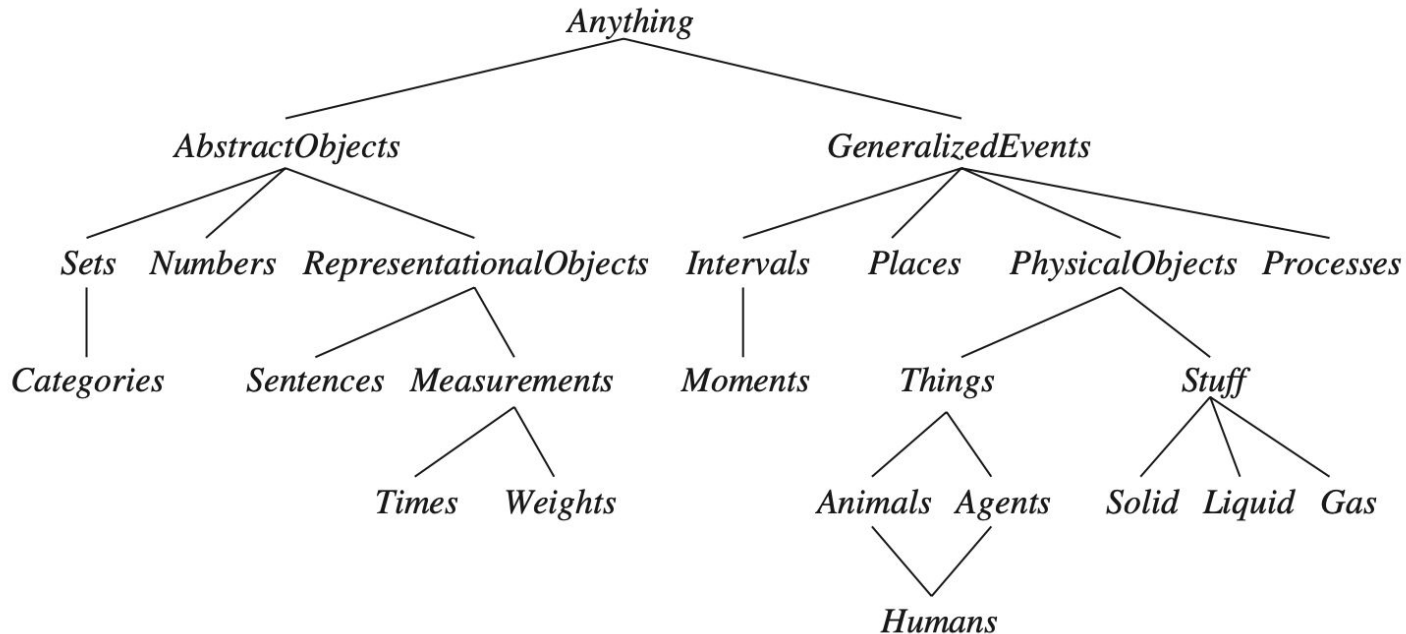
- > We need to create representations, concentrating on general concepts: Events, Time, Physical Objects, and Beliefs that occur in many different domains.

An **ontology** aims at organising everything in the world into a hierarchy of categories. More precisely, “An ontology is explicit specification of a conceptualization”.

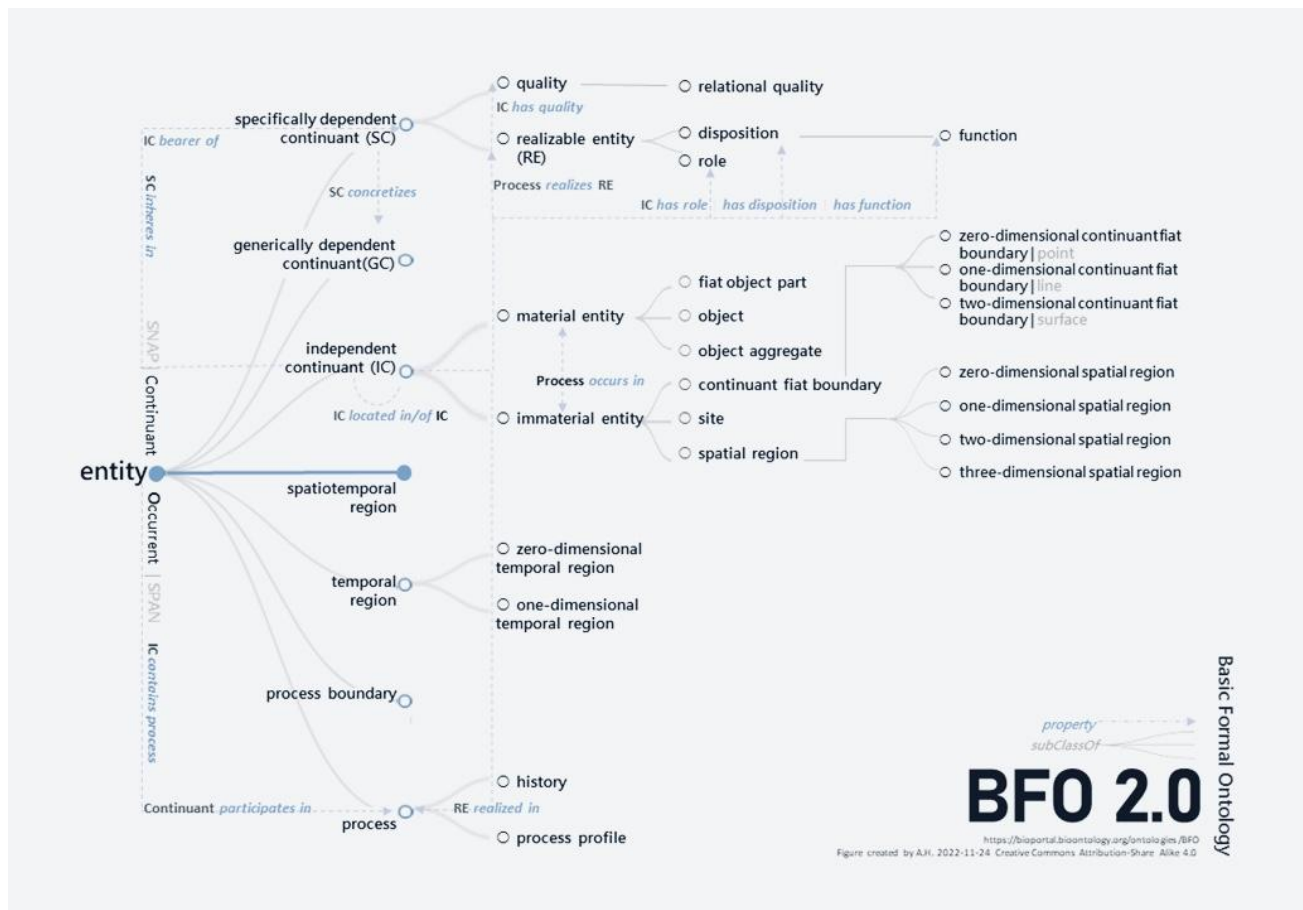
Representing abstract concepts is called **ontological engineering**.

# Upper ontology of the world

The **upper ontology** is a general framework of concepts, it aims to capture universal and abstract concepts that are common to a wide range of knowledge areas



# Upper ontology – BFO



# Upper ontology characteristics

**Standardization**, a common set of terms and relationships that can be adopted uniformly across different applications and domains

**Interoperability**, enable the exchange of information between diverse sources.

**Reusability**, concepts can be reused in various domains, (reducing the need to reinvent the wheel for common, foundational concepts).

## Categories and objects

*Although interaction with the world takes place at the level of individual objects, much reasoning takes place at the level of categories.*

For example, we interact with a cat named Lilly as an **individual object**, but when we want to buy food we deal with her as a **category**.

# Inheritance

*A class of objects (category) **inherit** the properties and behaviors of its superclass*

For example:

- all instances of the category Food are edible;
- Fruit is a subclass of Food and Apples is a subclass of Fruit,
- **we infer that every apple is edible** (the individual apples inherit the property of edibility, following their membership in the Food category).

# Objects and categories In FOL

An object is a member of a category.

$BB9 \in \textit{Basketballs}$

A category is a subclass of another category.

$\textit{Basketballs} \subset \textit{Balls}$

All members of a category have some properties.

$(x \in \textit{Basketballs}) \Rightarrow \textit{Spherical}(x)$

Members of a category can be recognized by some properties.

$\textit{Orange}(x) \wedge \textit{Round}(x) \wedge \textit{Diameter}(x)$



# Objects and categories In FOL

An object is a member of a category.

$BB9 \in \textit{Basketballs}$

A category is a subclass of another category.

$\textit{Basketballs} \subset \textit{Balls}$

All members of a category have some properties.

$(x \in \textit{Basketballs}) \Rightarrow \textit{Spherical}(x)$

Members of a category can be recognized by some properties.

$\textit{Orange}(x) \wedge \textit{Round}(x) \wedge \textit{Diameter}(x)$

> *might be an orange ball for another sport ?!*

# Sets

# Sets

A set is a well-defined collection of objects (namely, elements of the set)

Formally, a set can be defined as: {1,4,8,3,6}

If  $x$  is an element of set  $A$  then:  $x \in A$

If  $x$  is not an element of set  $A$  then:  $x \notin A$

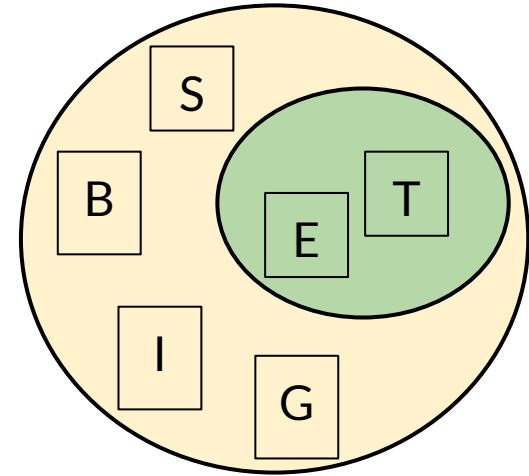
# Types of sets

If every element of a set B is also an element of another set A, then we say B is a subset of set A

Given:  $A = \{B, I, G, S, E, T\}$  and  $B = \{E, T\}$

Then B is a subset of A:  $B \subseteq A$

Two sets A and B are equal ( $A = B$ )  
if they have the same type and number of elements



$$B \subseteq A \text{ and } A \subseteq B$$

# Disjoint sets

Two sets A and B are **disjoint**, if they do not have any element in common

## Different elements:

A: {1, 2, 3}

B: {4, 5, 6}

## Type of elements:

A: {A, B, C}

B: {1, 2, 3, 4}

# Operations on sets: union and intersection

The **union** combines two sets into a new set containing all unique elements from both sets.

$$A \cup B$$

$$\{1, 2, 3, 4\} \cup \{2, 5\} = \{1, 2, 3, 4, 5\}$$

The **intersection** between two sets includes only elements that are present in both sets.

$$A \cap B$$

$$\{1, 2, 3, 4\} \cap \{2, 5\} = \{2\}$$

## Example

Five friends are planning a movie night and want to choose a film that everyone will enjoy. Each friend has their own preferred genres—comedy, action, drama, adventure, and horror. Using set theory, determine the best movie options that align with everyone's tastes!

# Example

Five friends are planning a movie night and want to choose a film that everyone will enjoy. Each friend has their own preferred genres—comedy, action, drama, adventure, and horror. Using set theory, determine the best movie options that align with everyone's tastes!

$F_1 = \{ \langle \text{genres} \rangle \}$

$F_2 = \{ \langle \text{genres} \rangle \}$

$F_3 = \{ \langle \text{genres} \rangle \}$

$F_4 = \{ \langle \text{genres} \rangle \}$

$F_5 = \{ \langle \text{genres} \rangle \}$



# Example

Five friends are planning a movie night and want to choose a film that everyone will enjoy. Each friend has their own preferred genres—comedy, action, drama, adventure, and horror. Using set theory, determine the best movie options that align with everyone's tastes!

$F_1 = \{ \langle \text{genres} \rangle \}$

$F_2 = \{ \langle \text{genres} \rangle \}$

$F_3 = \{ \langle \text{genres} \rangle \}$

$F_4 = \{ \langle \text{genres} \rangle \}$

$F_5 = \{ \langle \text{genres} \rangle \}$

$F_1 \cap F_2 \cap F_3 \cap F_4 \cap$   
 $F_5$

# Example

Five friends are planning a movie night and want to choose a film that everyone will enjoy. Each friend has their own preferred genres—comedy, action, drama, adventure, and horror. Using set theory, determine the best movie options that align with everyone's tastes!

$F_1 = \{ \text{"comedy"}, \text{"action"}, \text{"horror"} \}$

$F_2 = \{ \text{"horror"} \}$

$F_3 = \{ \text{"comedy"}, \text{"horror"} \}$

$F_4 = \{ \text{"drama"}, \text{"horror"}, \text{"action"} \}$

$F_5 = \{ \text{"horror"}, \text{"action"}, \text{"adventure"} \}$

$F_1 \cap F_2 \cap F_3 \cap F_4 \cap F_5 =$   
 $\{ \text{"horror"} \}$

# Example

Five friends are planning a movie night and want to choose a film that everyone will enjoy. Each friend has their own preferred genres—comedy, action, drama, adventure, and horror. Using set theory, determine the best movie options that align with everyone's tastes!

$F_1 = \{ \text{"comedy"}, \text{"action"}, \text{"horror"} \}$

$F_2 = \{ \text{"comedy"}, \text{"drama"} \}$

$F_3 = \{ \text{"comedy"}, \text{"horror"} \}$

$F_4 = \{ \text{"drama"}, \text{"horror"}, \text{"action"} \}$

$F_5 = \{ \text{"horror"}, \text{"action"}, \text{"adventure"} \}$

What if the intersection  
is empty?

# Example

Five friends are planning a movie night and want to choose a film that everyone will enjoy. Each friend has their own preferred genres—comedy, action, drama, adventure, and horror. Using set theory, determine the best movie options that align with everyone's tastes!

$F_1 = \{ \text{"comedy"}, \text{"action"}, \text{"horror"} \}$

$F_2 = \{ \text{"comedy"}, \text{"drama"} \}$

$F_3 = \{ \text{"comedy"}, \text{"horror"} \}$

$F_4 = \{ \text{"drama"}, \text{"horror"}, \text{"action"} \}$

$F_5 = \{ \text{"horror"}, \text{"action"}, \text{"adventure"} \}$

We follow the majority!

# Example

Five friends are planning a movie night and want to choose a film that everyone will enjoy. Each friend has their own preferred genres—comedy, action, drama, adventure, and horror. Using set theory, determine the best movie options that align with everyone's tastes!

**Calculate all possible intersections using 4 sets**

$F_1 = \{ \text{"comedy"}, \text{"action"}, \text{"horror"} \}$

$F_2 = \{ \text{"comedy"}, \text{"drama"} \}$

$F_3 = \{ \text{"comedy"}, \text{"horror"} \}$

$F_4 = \{ \text{"drama"}, \text{"horror"}, \text{"action"} \}$

$F_5 = \{ \text{"horror"}, \text{"action"}, \text{"adventure"} \}$

$F_1 \cap F_2 \cap F_3 \cap F_4 = \{ \}$

$F_2 \cap F_3 \cap F_4 \cap F_5 = \{ \}$

**$F_3 \cap F_4 \cap F_5 \cap F_1 = \{ \text{"horror"} \}$**

$F_4 \cap F_5 \cap F_1 \cap F_2 = \{ \}$

$F_5 \cap F_1 \cap F_2 \cap F_3 = \{ \}$

**At least 4 people like "horror"**

# Example

Five friends are planning a movie night and want to choose a film that everyone will enjoy. Each friend has their own preferred genres—comedy, action, drama, adventure, and horror. Using set theory, determine the best movie options that align with everyone's tastes!

$F_1 = \{ \text{"comedy"}, \text{"action"} \}$

$F_2 = \{ \text{"drama"} \}$

$F_3 = \{ \text{"comedy"}, \text{"horror"} \}$

$F_4 = \{ \text{"drama"}, \text{"horror"}, \text{"action"} \}$

$F_5 = \{ \text{"horror"}, \text{"action"}, \text{"adventure"} \}$

What if all possible intersections using 4 sets is NULL ?

Lets try by combining **3** sets: it's still the majority!

# Example

Five friends are planning a movie night and want to choose a film that everyone will enjoy. Each friend has their own preferred genres—comedy, action, drama, adventure, and horror. Using set theory, determine the best movie options that align with everyone's tastes!

$F_1 = \{ \text{"comedy"}, \text{"action"} \}$

$F_2 = \{ \text{"drama"} \}$

$F_3 = \{ \text{"comedy"}, \text{"horror"} \}$

$F_4 = \{ \text{"drama"}, \text{"horror"}, \text{"action"} \}$

$F_5 = \{ \text{"horror"}, \text{"action"}, \text{"adventure"} \}$

$$F_1 \cap F_2 \cap F_3 = \{\}$$

$$F_2 \cap F_3 \cap F_4 = \{\}$$

$$F_3 \cap F_4 \cap F_5 = \{ \text{"horror"} \}$$

$$F_4 \cap F_5 \cap F_1 = \{ \text{"action"} \}$$

...

*(the rest is all empty)*





# Relations between categories

*We want to state relations between categories that are not subclasses of each other.*

For example:

→ if we just say that *Undergraduates* and *GraduateStudents* are subclasses of *Student*, then we have not said that an undergraduate cannot also be a graduate student.

Two or more categories:

- are **disjoint** if they have no members in common
- form an **exhaustive decomposition** of another category C – if each object of C belongs to at least one of these categories
- form a **partition** if all categories in the **exhaustive decomposition** are disjoint

# Relations between categories – example

## Disjoint

A = *Animals*,

B = *Trees*

## Exhaustive Decomposition of *Continents*:

A = {Africa, Antarctica, Asia, Europe, North America}

B = {South America, Oceania}

## Partition

A = {Africa, Antarctica, Asia, Europe, North America, South America, Oceania}

B = *Continents*

# Physical composition

*The idea that one object can be part of another is a familiar one*

For example, a nose is part of the head, Italy is part of Europe, and a chapter is part of a book

The **PartOf** relation is used to say that one thing is part of another. Objects can be grouped into *PartOf* hierarchies, (to indicate the Subset hierarchy)

For example:

*PartOf (Bologna, Italy)*

*PartOf (Italy, Europe)*

*PartOf (Europe, Earth)*

# Reasoning Systems for Categories

Family Systems specially designed for organizing and reasoning with categories, are:

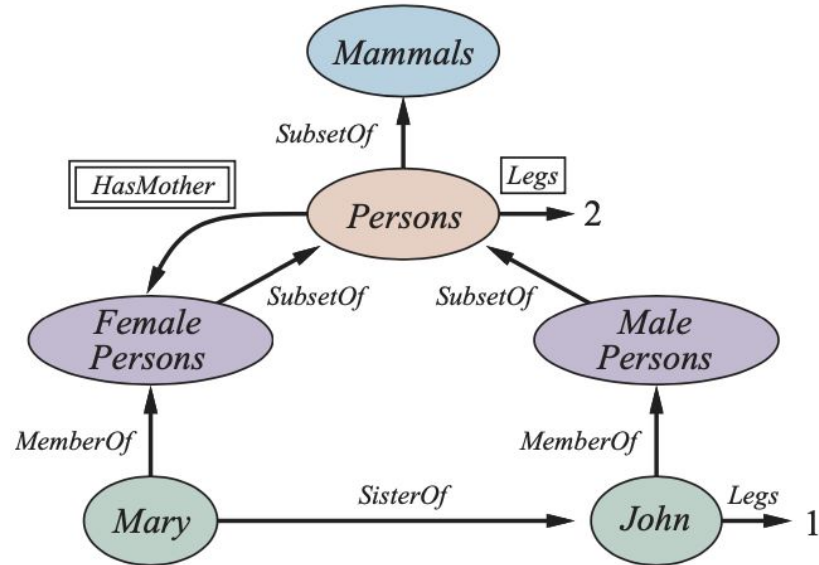
- **Semantic networks** providing a graphical aids for visualizing a knowledge base
- **Description logics** providing a formal language for constructing and combining category definitions and efficient algorithms for deciding subset and superset relationships between categories.

# Semantic networks

*A typical graphical notation displays object or category names in ovals or boxes, and connects them with labeled links*

The single-boxed link is used to assert properties of every member of a category.

The double-boxed link is used to assert a relation between an **object person and his mother (not the entire category)**.



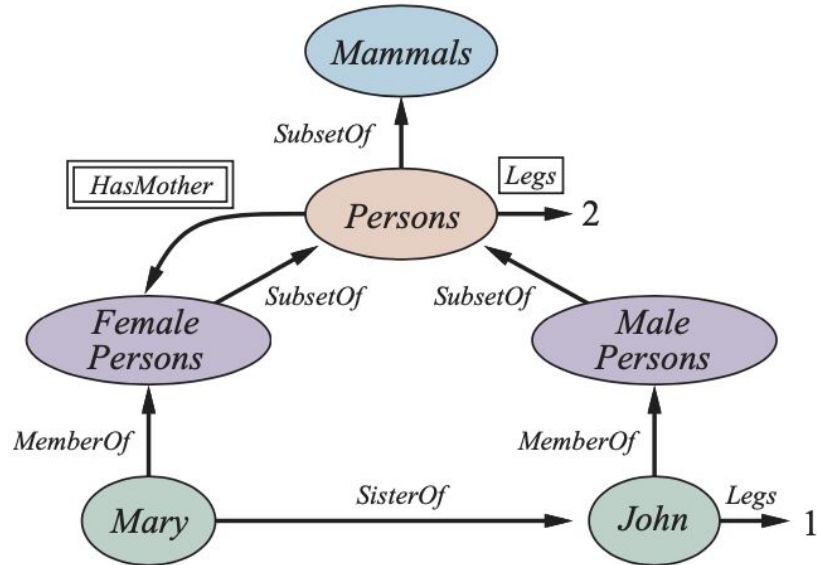
# Semantic networks

*A typical graphical notation displays object or category names in ovals or boxes, and connects them with labeled links*

The double-boxed link is used to assert a relation between **an object person and his mother (not the entire category)**.

Can we draw a *HasMother* link with no box from *Persons* to *FemalePersons*?

NO: it's not a property



# Semantic networks – multiple inheritance

*What if a category is a subset of more than one other category; what does it inherit?*

This is called **multiple inheritance**.

> In such cases, the inheritance algorithm might find two or more conflicting values.

For this reason, multiple inheritance is banned in some object-oriented programming languages, e.g. JAVA

*Example,  
if **Bats** is a subclass of both **Birds** and **Mammals** and both have the property “**has legs**”, which one should he inherit?*

# Description logics

**Description logics** are notations that are designed to make it easier to describe definitions and properties of categories.

**Description logic** systems evolved from semantic networks in response to pressure to formalize what the networks mean while retaining the emphasis on taxonomic structure as an organizing principle.

**Inference** tasks for description logics are mainly:

- **subsumption** (checking if one category is a subset of another by comparing their definitions)
- **classification** (checking whether an object belongs to a category).
- **consistency of a category definition**—whether the membership criteria are logically satisfiable.



# Description logics: subsumption

**Subsumption** checks whether a category (subclass) is a subset of another (superclass)

→ All objects belonging to the subclass also belong to the superclass.

For example:

$\text{Dog} \sqsubseteq \text{Pet} \sqcap \text{HasFur}$

$\text{Pet} \sqsubseteq \text{Mammal} \sqcap \text{NonHuman}$

Subsumption is used to check if Dog is a subset of Pet, and it compares the definitions of both categories.

→ In this case, all dogs are pets with fur, and all pets are non-human mammals. Therefore, Dog is a subset of Pet.

# Description logics: classification

**Classification** checks whether an object (instance) belongs to a category (class).

→ Verifies whether the properties of the instance satisfy the membership criteria of the class.

For example:

- Let's consider the instance **Fido**, which has the properties fur and being a pet.
- Using classification we determine whether Fido belongs to the **Dog** class by verifying whether its properties match the definition of a dog.
- If we define the **Dog** class as "a pet with fur", and Fido satisfies these conditions, then Fido can be classified as a Dog.

## Description logics: consistency

**Consistency** checks if the definition of a category is logically satisfiable.

→ Verifies if there exist objects that can simultaneously satisfy all the membership criteria of the category.

For example:

Let's consider the following definition:

$\text{Bird} \equiv \text{Animal} \sqcap \text{HasWings} \sqcap \text{CanFly}$

The definition is inconsistent because it implies that all birds must have wings and the ability to fly. However, there are flightless birds (e.g., ostriches, penguins) that do not possess the ability to fly.

# Description logics – example

CLASSIC is a description logic of objects in terms of their relations to other known objects, and their level of intensional structure.

*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* | ...

# Exercises

# Exercise

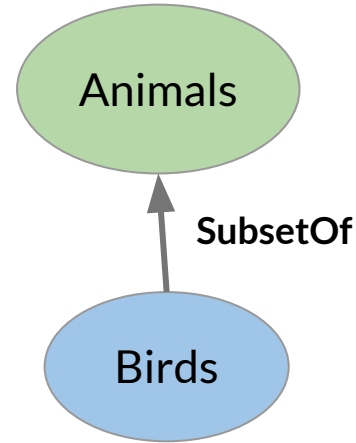
Represent the following sentences into a semantic network.

- Birds are animals.
- Birds can fly and lay eggs.
- Albatros is a bird.
- Donald is a bird.
- Tracy is an albatros.

# Exercise

Represent the following sentences into a semantic network.

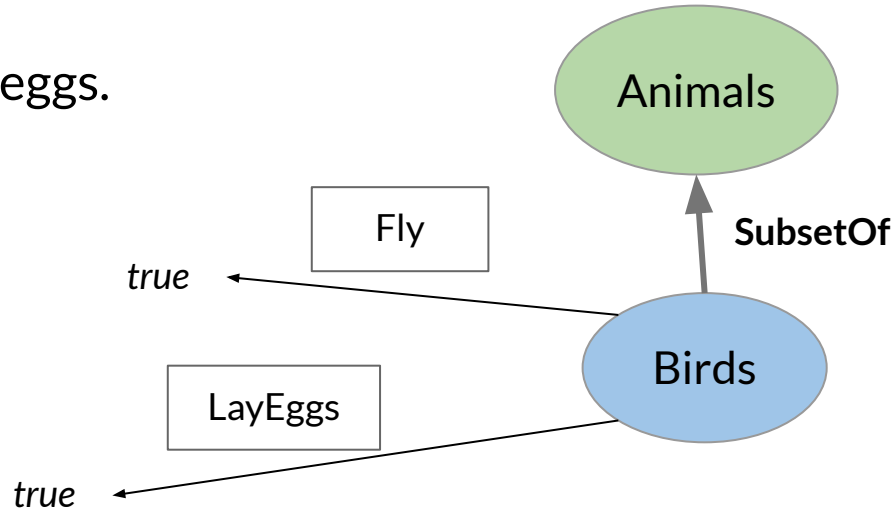
- Birds are animals.
- Birds can fly and lay eggs.
- Albatros is a bird.
- Donald is a bird.
- Tracy is an albatros.



# Exercise

Represent the following sentences into a semantic network.

- Birds are animals.
- Birds can fly and lay eggs.
- Albatros is a bird.
- Donald is a bird.
- Tracy is an albatros.

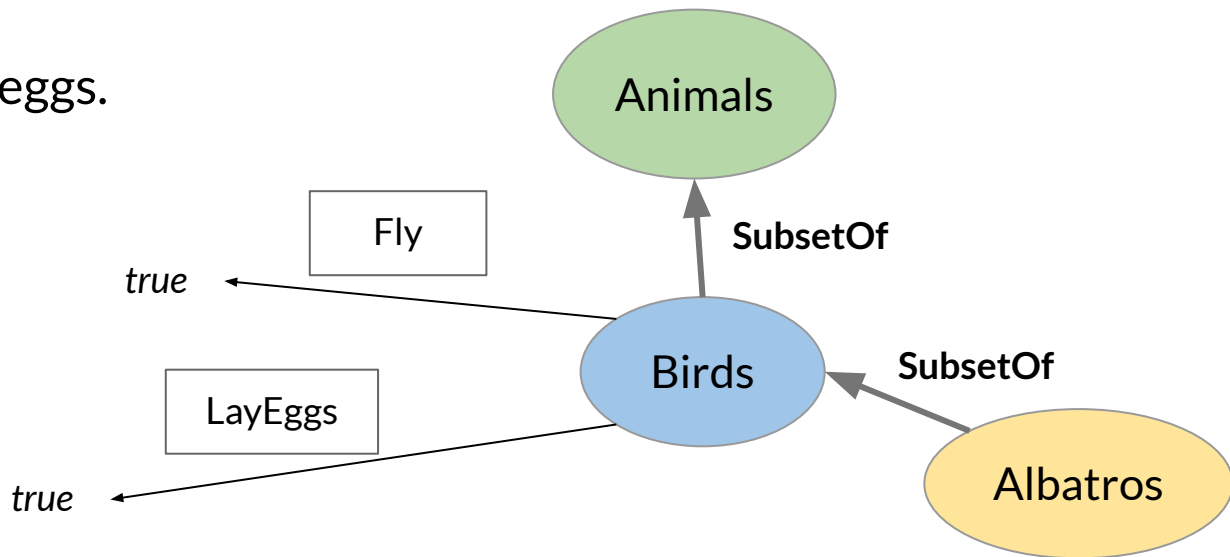




# Exercise

Represent the following sentences into a semantic network.

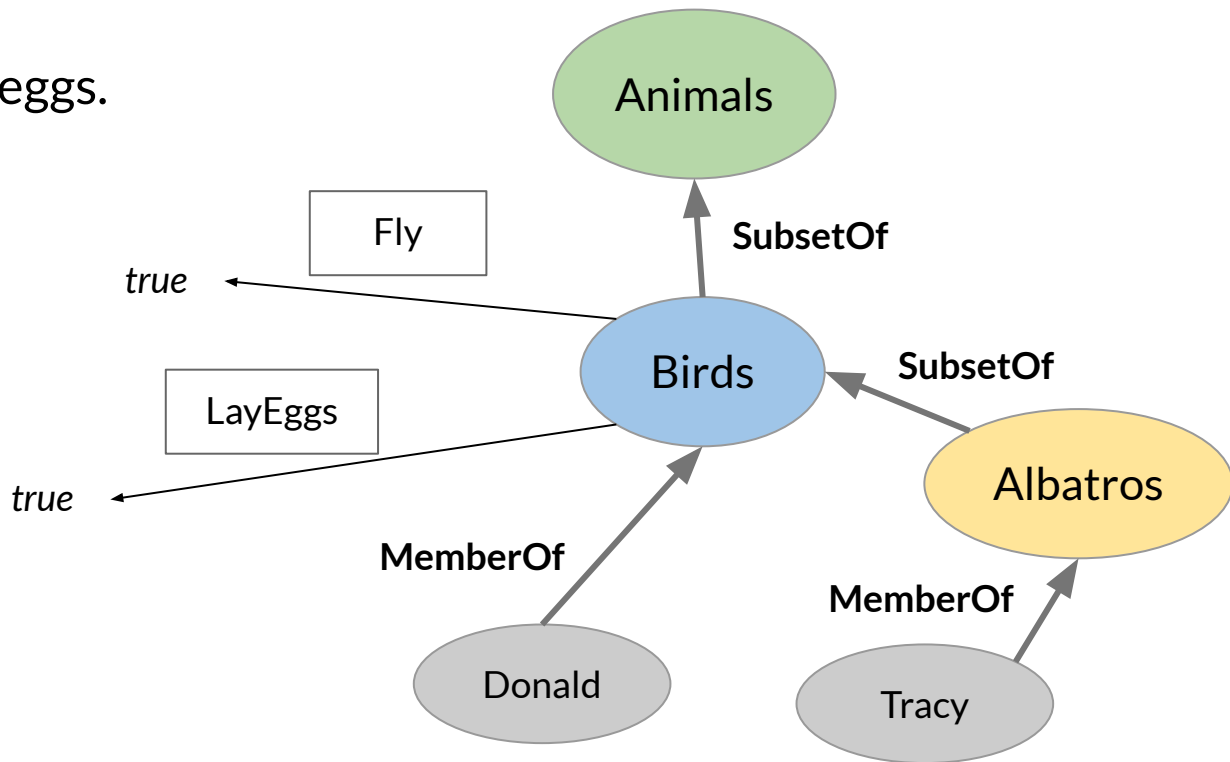
- Birds are animals.
- Birds can fly and lay eggs.
- Albatros is a bird.
- Donald is a bird.
- Tracy is an albatros.



# Exercise

Represent the following sentences into a semantic network.

- Birds are animals.
- Birds can fly and lay eggs.
- Albatros is a bird.
- Donald is a bird.
- Tracy is an albatros.



# Object-oriented programming

Object-oriented programming is a programming paradigm that organizes software around objects, which are instances of classes representing categories or types

In the Object Oriented Programming world, we can classify objects relations into two main relations:

- Is-A: describes **inheritance**,  
e.g., *Ferrari* is-A *Car*
- Has-A: describes **composition**,  
e.g., *Car* Has-A *Engine*

