FUNDAMENTAL OF A.I Module 2

Nell’esami ci saranno 2 esercizi su Prolog, uno semplice e l’altro sarà un problema meta-interpeter

**A common language: First Order Logic**

*Introduction*

Natural language is subject, by its own nature, to ambiguity

In this course we will discuss about knowledge and reasoning: the risk of ambiguities is higher

We need to agree to a language that is free of ambiguity

Choice: First Order Logic, or better saying, logic in general

*Lesson*

We need a notation for representing reasoning: symbols, the act of reasoning, interpretations.

As a notation we can use symbols:

- Symbols can be anything, for us they will be signs drawn on a piece of paper or a slide

– We will associate a symbol to a concept (more on this, later)

– We will use symbols together and create sentences

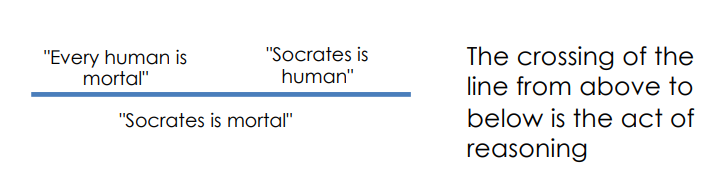
– Sentences, sometime, might be true or false

We will represent the "act of reasoning" in the following way:

– we will write the things that we know (or believe we know, or that we know they are true)

– a horizontal line

– below the line, the result of our reasoning act

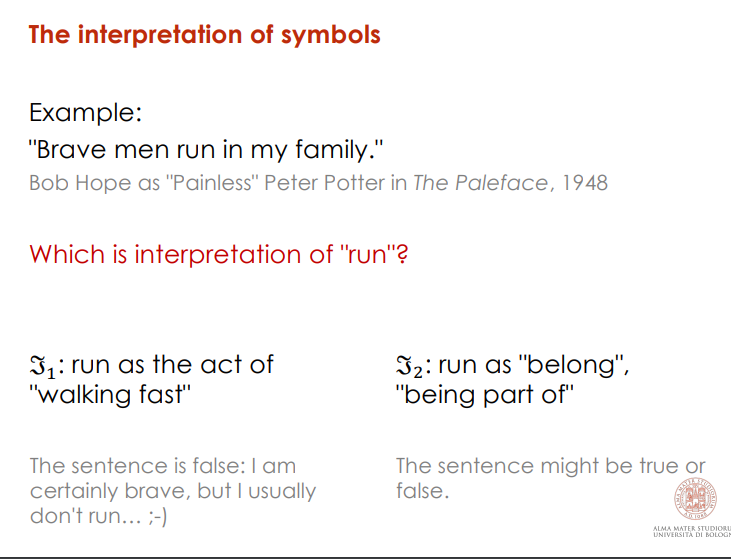


If anyone can have her/his own interpretation, what happens to the "correctness" of the reasoning?

An interpretation ℑ is a pair (D, I) where:

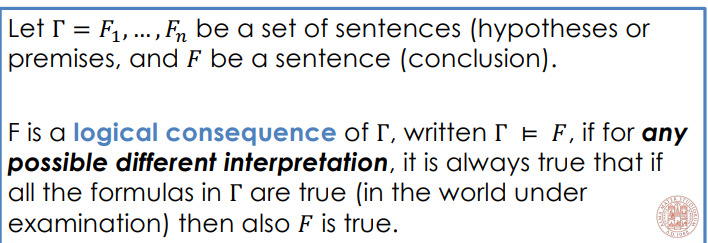
• D is any set of objects, called domain

• I is a mapping called the interpretation mapping, from the (non logical) symbols to elements and relations over D

Given a set of sentences, their truth value will change depending on the interpretation we will give to the symbols: there is a strict relation between the notion of interpretation and the truth value. Can we reason, then?

To avoid the problem of the many different interpretations of symbols, we will restrict:

• from the notion of truth (something more related to philosophy) to the notion of Logical Consequence



This is called automatation framework.

*Propositional Logic*

Propositional logic is the simplest logic, it illustrates basic ideas using symbols called propositions Pi

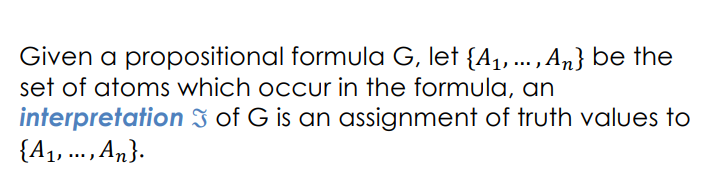
Each Pi is referred as an atom or atomic formula

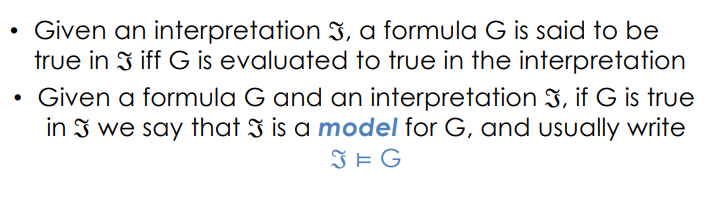
• Atoms Pi can be connected together by connectives: ∧ (and) , ∨ (or), → (implies), ¬(not), ↔ (double implication), ⊥ (false)

• A well-formed formula is:

– an atomic formula Pi

– if A and B are well-formed formulas, then A ∧ B, A ∨ B, A → B, and ¬A are well-formed formulas





Inconsistent is a sentence that we can never made it true

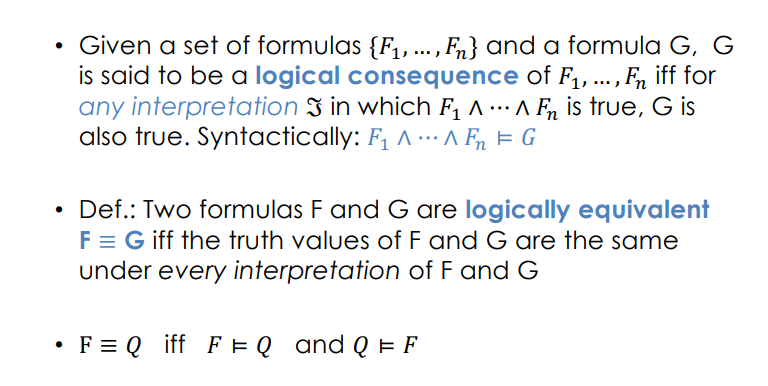
Invalid are those sentences that sometimes can be true sometimes are false

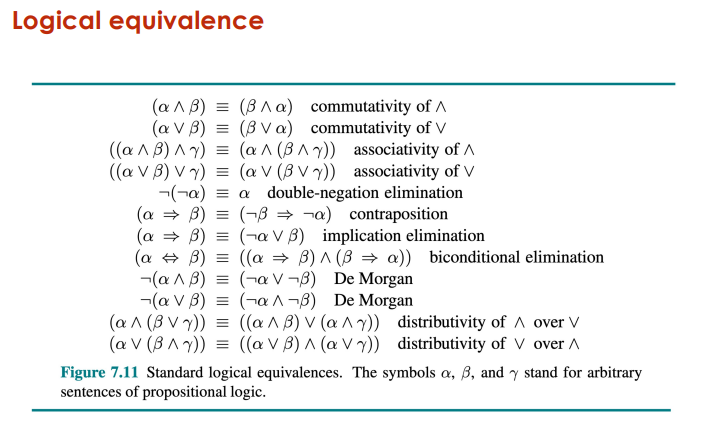
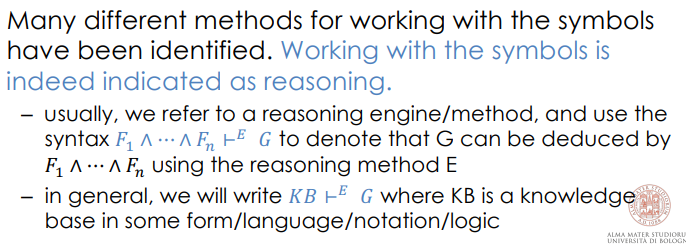
How to decide if a formula is valid or not?

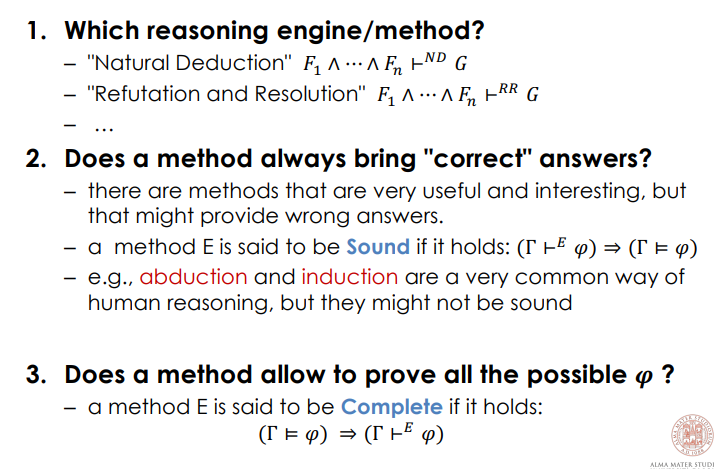
1. enumerate all the possible interpretations (the number of interpretations are finite)

2. look at the formula for each interpretation

Our goal is to decide if, given some formulas are true, a formula G is true as well







Abduction is used to make hypotetsys, the grass is wet, it is raining.

Esempi programmi in Porlog:

1)

q :- p. // (p -> q) :- == 🡨

p.

r :- q.

We start the program on prolog on the terminal and we will get

>> r.

true.

Ok next Step

2) P is true if we are able to write Hello World

p :- write(‘Hello World’). //this is called define predicate

We start the program on prolog on the terminal and we will get

>> Hello World

true.

3)

P(X) :- write(X).

We start the program on prolog on the terminal and we will get

>> p.

Error! Unknow parameter

>> P(wonderful)

wonderful

True.

4)

P(X, Output) :- Output is X+1.

We start the program on prolog on the terminal and we will get

>> p(5, Result).

Result = 6.

5) Fattoriale

% fact (PostNumber, Result).

fact(0, 1).

fact(PostNumber, Result) :- Pos1 is PosNumber -1,

fact(Pos1, Result1),

Result is PosNumber = Result1.

We start the program on prolog on the terminal and we will get

>> p(5, Result).

Result = 6.

Upper Ontologies

There are many different types of knowledge. However, there is a limited number of very high-level concepts that appear constantly -> Upper ontologies

An ontology is a formal, explicit description of a domain of interest.

Generally speaking, we (humans) tend to organize our knowledge through two notions:

• Categories

• Objects (belonging to the categories)

There are categories (staff vs. professors)

• Categories can be more or less general (person vs. professor)

• Categories can be organized hierarchically

• Objects can be instances of categories

• Objects can belong to more categories at the same time

An upper ontology is an ontology that focuses on the largest, more general, more high level, most general purpose domain. (Example: Person)

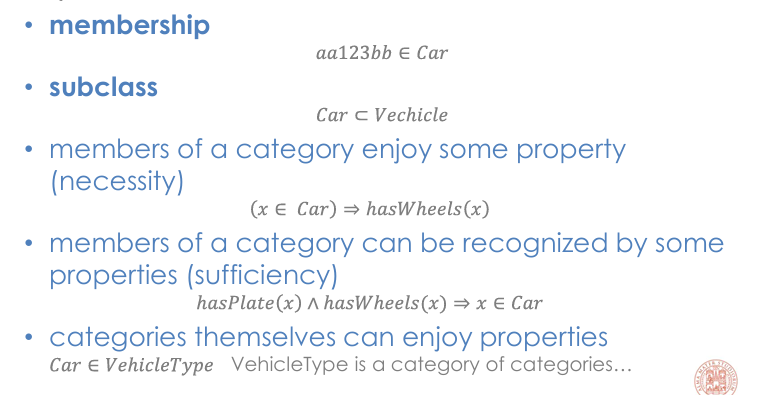
Categories and objects: how to represent them?

Objects: it is safe to assume there exists always a way to identify each of them, a unique name

n FOL, two possible alternatives:

1. categories as predicates e.g.: Car is a predicate that represents the category of cars; Car(aa123bb) means that the vehicle with plate aa123bb is a car
2. through reification: categories as objects as well

E.g.: Member(aa123bb, Car)



Proprieties of Categories:

**Disjointness**: A set of categories were objects can be only in one of these categories.

Example: Linneus stated that every animal could not be a vegetable, and the opposite. We would say that Animals and Vegetables are two disjointed categories,

**Exhaustive Decomposition:** Subcategories might "cover" all the possible instances of the parent category.

Every one in this classroom is a student, or a professor (or both). "Student" and "Professor" categories provide an exhaustive decomposition of the category "People in this classroom"

**Partition**: If a category can be decomposed in more categories such that:

• they form an exhaustive decomposition

• they are disjoint

Then we have a partition

All the people in the classroom wear glasses or not

Objects can be made of other objects… the relation between an object and its parts is called meronymy: some objects (meronyms) are part of a whole (holonym).

E.g.: A car is composed of an engine, four wheels, five seats,

Is there any structural relation between the parts, and between the parts and the whole? (other than the composition itself?)

• Answer yes: the relation is usually named PartOf

E.g.: PartOf(engine123,car\_aa123bc), PartOf(cylinder1345, engine123)

• Answer no: the relation is usually named BunchOf

E.g.: BunchOf({Bread1, Bread2, Bread3})

There is a distiction in Object:

* Things
* Stuff

Which one to use?

A solution comes when looking at the properties

• some properties are referred to the substance of the object

E.g.: the water boils at 100°C Such properties are named intrinsic, and they are retained even when some division is applied

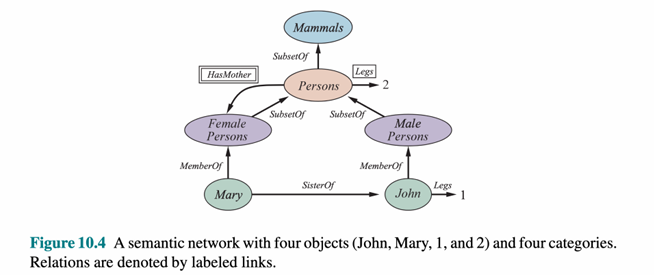
• some properties refer to the object, and to the object structure relation with its parts

E.g.: the weight of professor's car change if we take out a wheel Such properties are named extrinsic

• a category of objects that will include in its definition only intrinsic properties is a substance; Stuff is the most general substance category

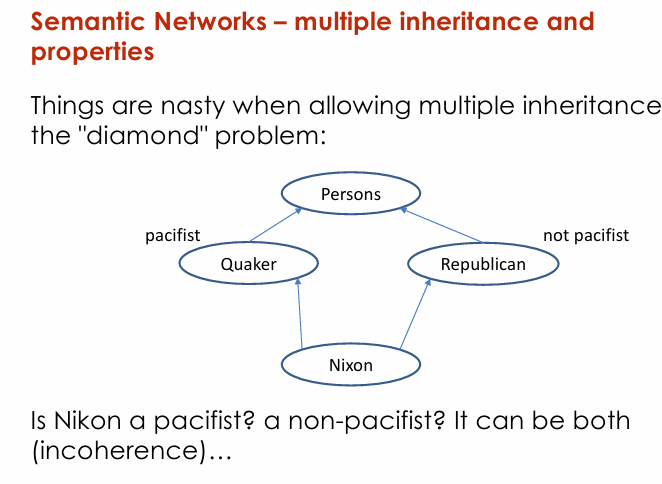
• a category of objects that will include in its definition any extrinsic property is a count noun; Thing is the most general object category

**Semantic Networks**



Categories and Objects represented with the same symbol…

Four different types of links, with the same graphical representation



It is a problem because with this solution we can provw whatever also stuff that are not true.

The trend right now for OOP Languages is to not allow multiple inheritance

Frames

Intuitively: a frame is a piece of knowledge that describes an object in terms of its properties

• it has a (unique) name

• each property is represented through couples slot – filler

(toronto

<:Instance-of City>

<Population 4.5M>

)

In the Frames proposal, an object belong to a category if it is similar enough to some typical members of the category, named prototypes.

Introduction to Descritpion Logic

Objects have parts, sometimes in multiples. The relationships among an object’s parts is essential to its being considered a member of a category.

Description Logic is a(family of) logic that focus on the description of the terms