## What is the difference between deductive inferences and commonsense inferences?

* Deductive inferences: We agree on the meaning of connectives between prepositions (ex: *and, or, not, if/then*). If the premises are true, **the consequences must be true**
* Commonsense inferences: We agree on the meaning of connectives between prepositions (ex: *and, or, not, if/then*). If the premises are true, **the consequences are justified**

## What are knowledge graph

Are a way to store entities that are connected by relations. They can be processed efficiently and offer a convenient abstraction

## What is the difference between passive applications and active applications?

* Passive applications: support exploratory search. Let users browse through the data before formulating a precise query
* Active applications: push content to the user, predict what the user is interested in. Like recommender systems

## How is a knowledge graph made?

A knowledge graph is a quintuple: <*E, L, T, P, A >*

* E: a set of **entity id** symbols (= constants)
* L: a set of **literals** strings, numbers… (= constants)
* T: a set of **type** symbols (= unary predicate symbols)
* P: a set of **relation** symbols (= binary predicate symbols)
* A: a set of axioms in a language

## What is an RDF?

RDF is a data model for representing data on the web. It is based on

* Triples: basic unit to organise the information
* Directed labelled graphs. Set of triples
* URIs - Unique Resource Identifier. A string of characters that uniquely identify a name or a resource on the internet

## What is a directed edge-labelled graph?

A directed edge-labelled graph is a tuple G = (V, E, L) where V ⊆ Con is a set of nodes, L ⊆ Con is a set of edge labels, and E ⊆ V ✕ L ✕ V is a set of edges

## What is a basic directed edge-labelled graph pattern?

We define a basic directed edge-labelled graph pattern as a tuple Q = (V, E, L) where V ⊆ Term is a set of node terms, L ⊆ Term is a set of edge terms, and E ⊆ V ✕ L ✕ V is a set of edges (triple patterns)

## What is an ontology and how many types exist?

Is a description of the concepts and relationships that can formally exist for an agent or a community of agents.

There are 3 types of ontologies:

1. **Tesauri/lexical ontologies**: dictionary for laguagem contains words and relations between words. Concepts are defined as sets of words that can have similar meaning in some contexts. Focus on word meaning
2. **Taxonomies**: classification structure for objects of a given domain. Usually hierarchies with tree-like structures. Focus on classification structures
3. **Axiomatic ontologies**: formal specification of a conceptualization with a language L, a set of logical axioms in L and a formal semantics that univocally specifies their meaning. Axiomatic ontologies organise knowledge by defining individuals, classes, relations. Focus on mathematical interpretation of the terms meaning

## What is an interpretation?

*I* = 〈Δi , ⋅i〉where:

* Δi is the domain of *I*
* ⋅i is an interpretation function that maps:
  + Each atomic concept C to a set Ci ⊆ Δi
  + Each atomic role R to a binary relation Ri ⊆ Δi ✕ Δi
  + Each individual name *a* to an element ai € Δi
  + ⋅i is extended recursively (complex concept/roles and axioms)

## Satisfiability, subsumption, equivalence, disjointness

* **Satisfiability**: a concept C is *satisfiable* with respect to T if there exists a model *I* of T such that Ci is nonempty. In this case we say that *I* is a model of C
* **Subsumption**: a concept C is *⊑* by a concept D with respect to T if Ci ⊆ Di for every model *I* of T. In this case we rite T ⊨ C ⊑ D
* **Equivalence**: two concepts C and D are *equivalent* with respect to T if Ci = Di for every model *I* of T. In this case we write T ⊨ C ≡ D
* **Disjointness**: two concepts C and D are *disjoint* with respect to T if Ci ∩ Di ≠ ø for every model *I* of T

## Predicate rule and models

A predicate rule is a predicate clause *P(t—) ← Q1(t1—) ,…, Qn(tn—)* such that all variables P(t—) appear in the body

A predicate knowledge base is a finite set of predicate rules

The intepretation *I* satisfies the rule *P(t—) ← Q1(t1—) ,…, Qn(tn—)* iff whenever *I* makes the body true, it also makes the head true

*I* is a model of the KB *K* iff it satisfies all rules in K

## What is a partial order relation?

R is a partial order relation iff R is

* Reflexive
* Antisymmetric
* Transitive