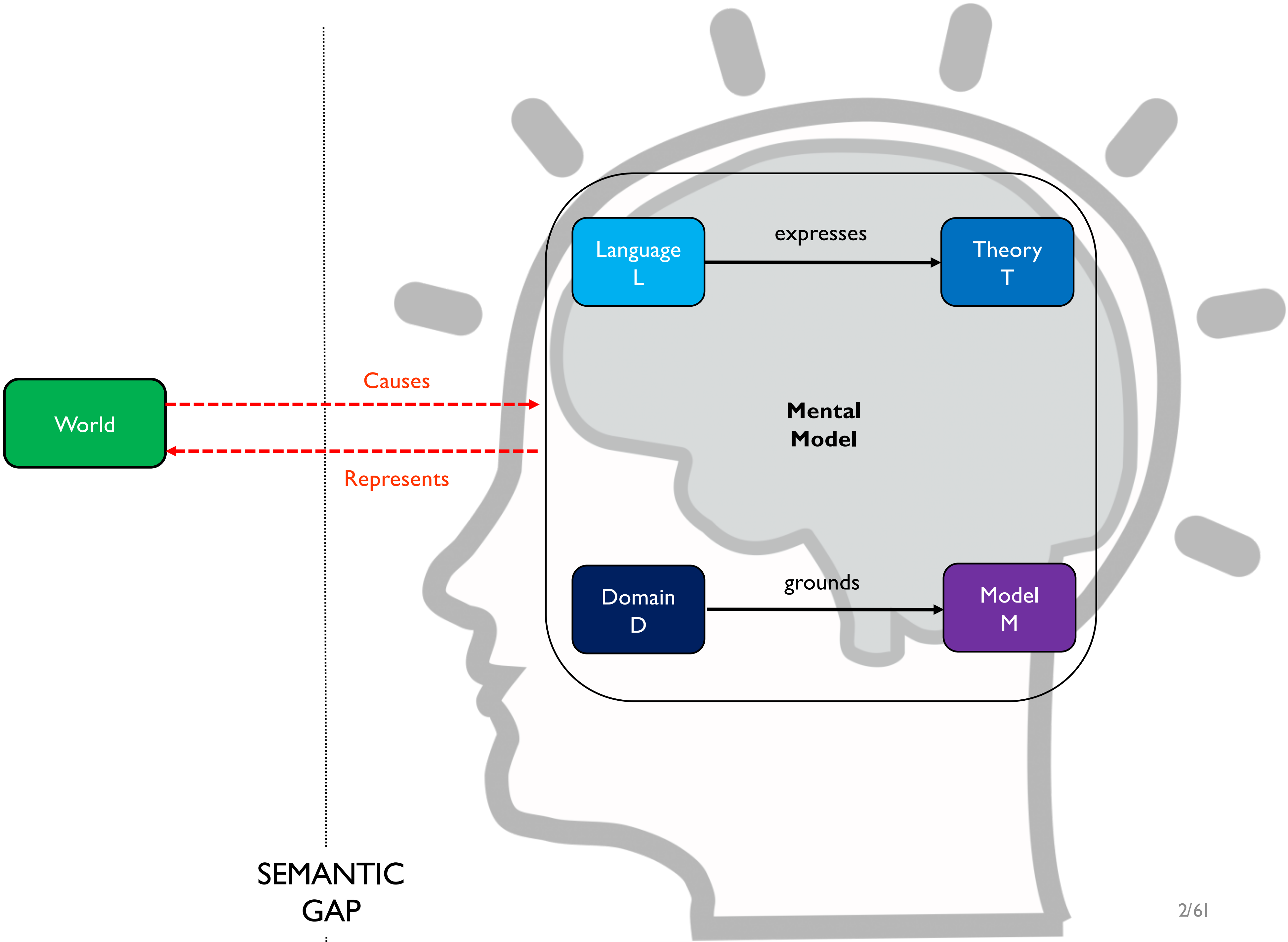
Mathematical Logics
Description Logic: Introduction

Fausto Giunchiglia and Mattia Fumagallli

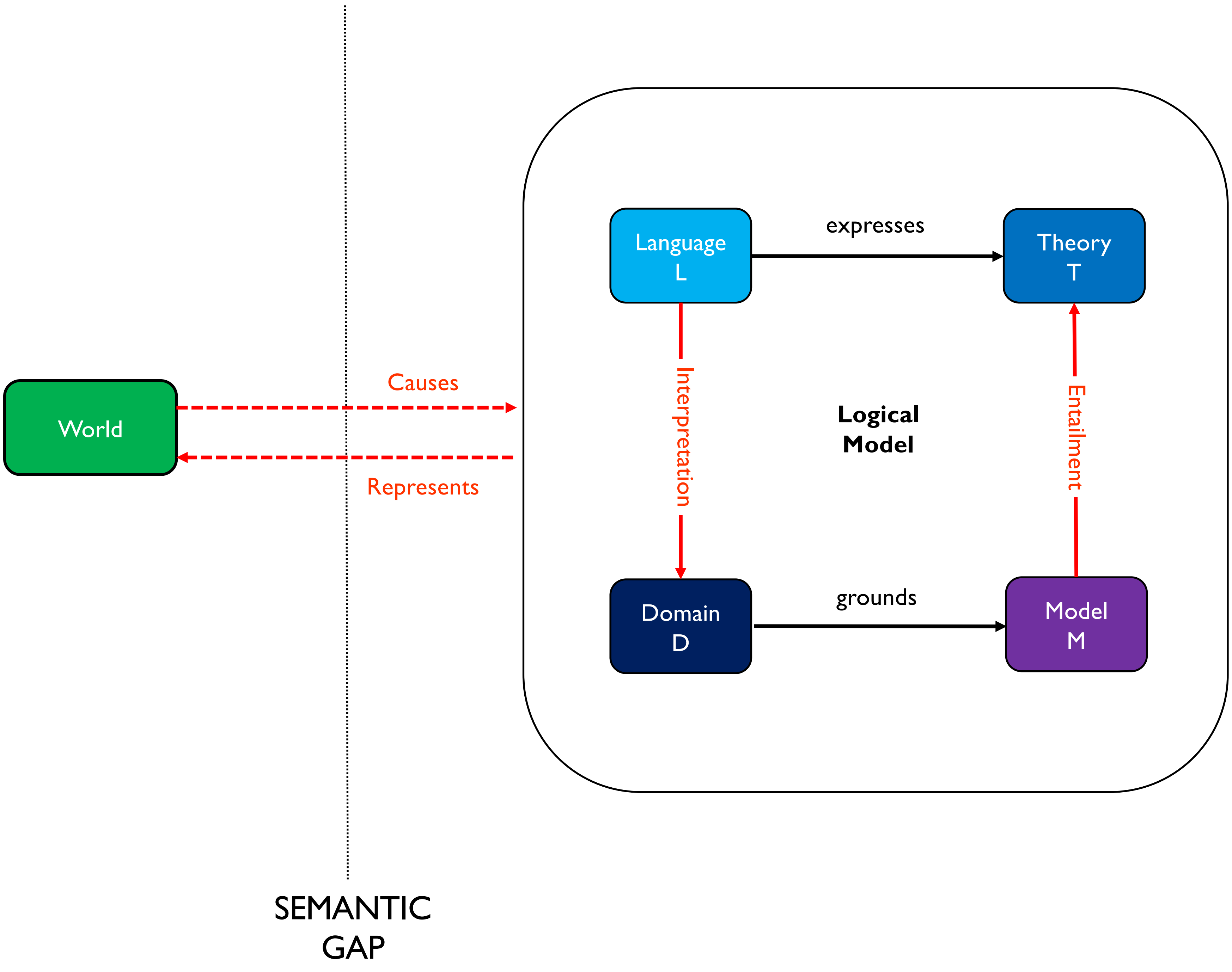
University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli

Mental Model



Logical Model



Logical Model

World

Logical Model

Language L

Domain D

Theory T

Model M

MONKEY	
/	Get
monkey#1	banana#2

BANANA	
/	Get^{-}
banana#2	monkey#2

L = "*Monkey, Above, Banana, Near, Get, monkey#1, banana#2, tree,* $\sqcap$, $\sqcup$, $\neg$, $\sqsubseteq$, $\exists$, $\forall$ …"

TBOX = "MonkeyGetBanana $\equiv$ Monkey $\sqcap$ $\forall$ Get.Banana"

D: {monkey#1, banana#2}

ABOX: "MonkeyGetBanana(monkey#1), Banana(banana#2)"

T,A $\models$ $\forall$ Get.Banana

SEMANTIC GAP

Where G informally means "Monkey gets banana"
Where $\exists$ stands for "Monkey actually gets Banana".

Language (Syntax)

- The syntax of ClassL is similar to PL
- Alphabet of symbols Σ_0

Σ_0

Descriptive

Logical

$\sqcap, \sqcup, \neg$

Constants

one proposition only

A, B, C …

Variables

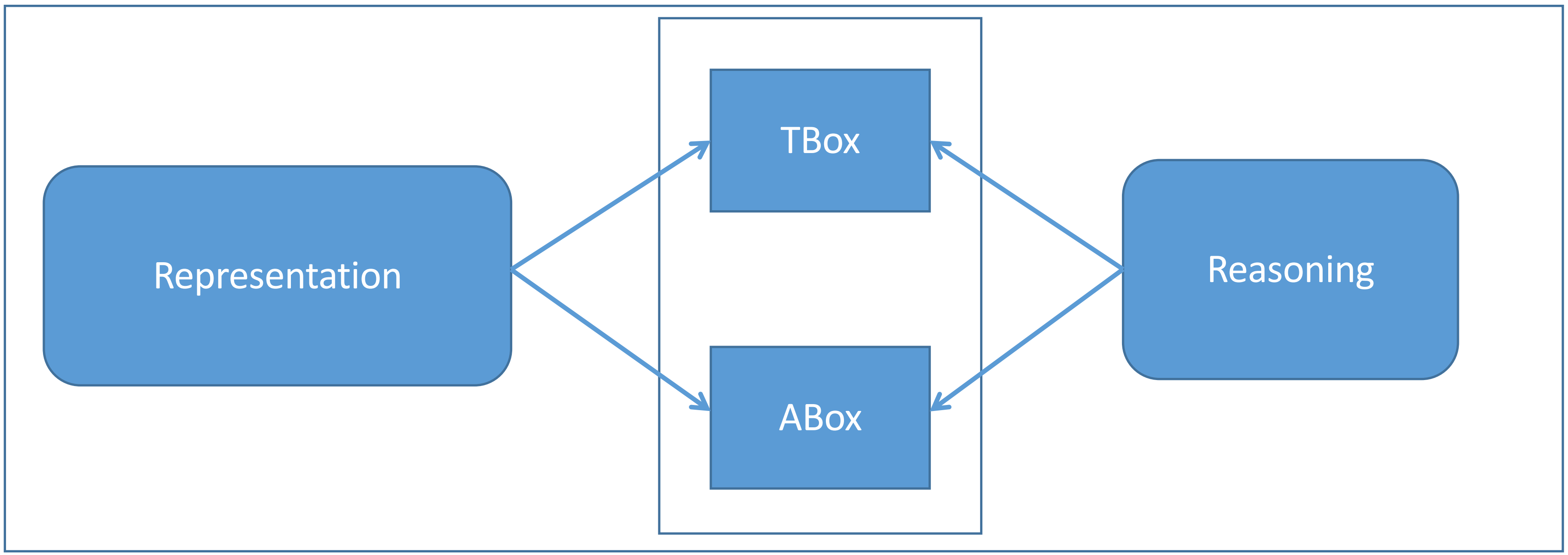
they can be substituted by any proposition or formula

P, Q, ψ …

NOTE: not only characters but also words (composed by several characters) like "monkey" are descriptive symbols

Overview

Description Logics (DLs) is a family of KR formalisms



Alphabet of symbols with two new symbols w.r.t. ClassL:

$\forall$R (value restriction)

$\exists$R (existential quantification)

R are atomic role names

Origins of Description Logics

Description Logics stem from early days knowledge representation formalisms (late '70s, early '80s):

Semantic Networks: graph-based formalism, used to represent the meaning of sentences.

Frame Systems: frames used to represent prototypical situations, antecedents of object-oriented formalisms.

Problems: no clear semantics, reasoning not well understood. Description Logics (a.k.a. Concept Languages, Terminological Languages) developed starting in the mid '80s, with the aim of providing semantics and inference techniques to knowledge representation system

What are Description Logics today?

In the modern view, description logics are a family of logics that allow to speak about a domain composed of a set of generic (pointwise) objects, organized in classes, and related one another via various binary relations. Abstractly, description logics allows to predicate about labeled directed graphs

vertexes represents real world objects

vertexes' labels represents qualities of objects

edges represents relations between (pairs of) objects

vertexes' labels represents the types of relations between objects.

Every piece of world that can be abstractly represented in terms of a labeled directed graph is a good candidate for being formalized by a DL.

What are Description Logics about?

Exercise

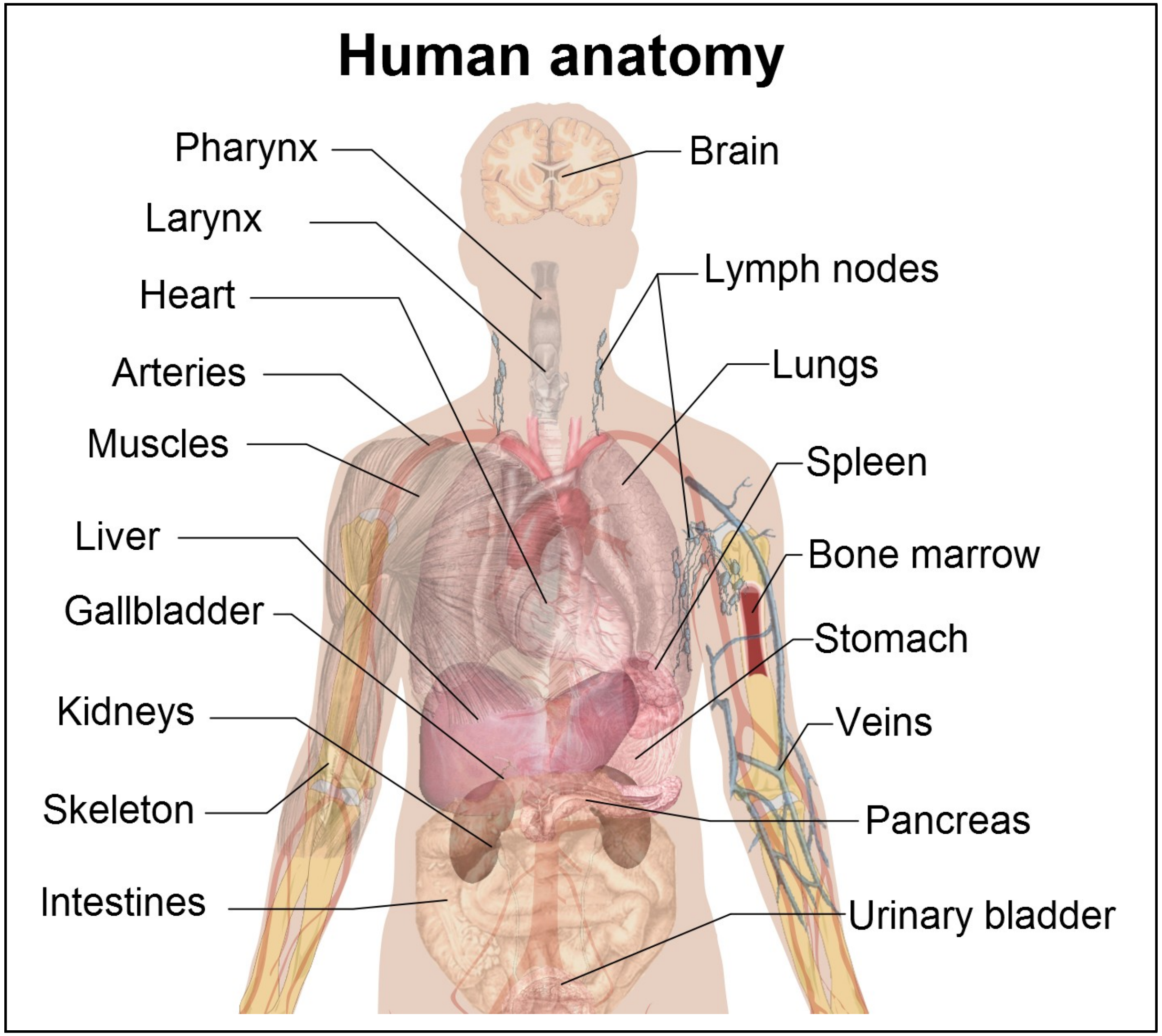
Represent Metro lines in Milan in a labelled directed graph

What are Description Logics about?

Exercise

Represent some aspects of Facebook as a labelled directed graph

What are Description Logics about?



Exercise

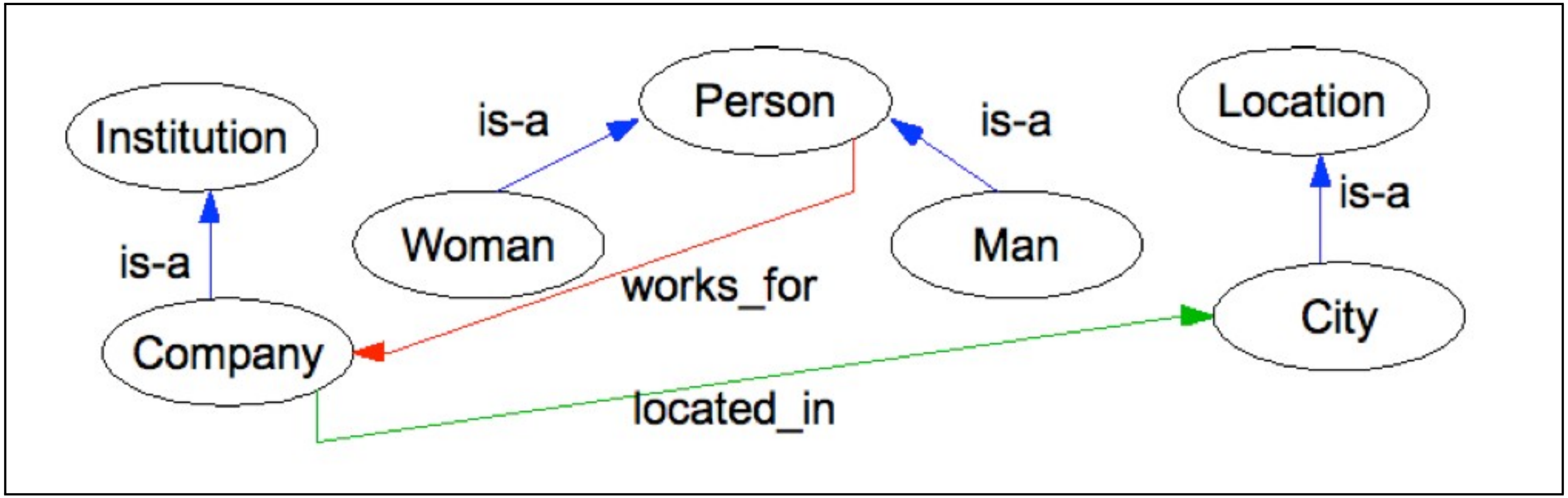
Represent some aspects of human anatomy as a labelled directed graph

What are Description Logics about?

Exercise

Represent some aspects of everyday life as a labelled directed graph

The everyday life example as a graph - intuition



- Family of logics designed for knowledge representation
- Allow to encode general knowledge (as above) as well as specific properties about objects (with individuals, e.g., *Mary*).

Ingredients of a Description Logic

A DL is characterized by:

A description language: how to form concepts and roles

$\text{Human} \sqcap \text{Male} \sqcap \exists \text{hasChild}.\top \sqcap \forall \text{hasChild}.(\text{Doctor} \sqcup \text{Lawyer})$

A mechanism to specify knowledge about concepts and roles (i.e., a TBox)

$$\mathcal{T} = \left\{ \begin{array}{l} \text{Father} \equiv \text{Human} \sqcap \text{Male} \sqcap \exists \text{hasChild}.\top \\ \text{HappyFather} \sqsubseteq \text{Father} \sqcap \forall \text{hasChild}.(\text{Doctor} \sqcup \text{Lawyer}) \\ \text{hasFather} \sqsubseteq \text{hasParent} \end{array} \right\}$$

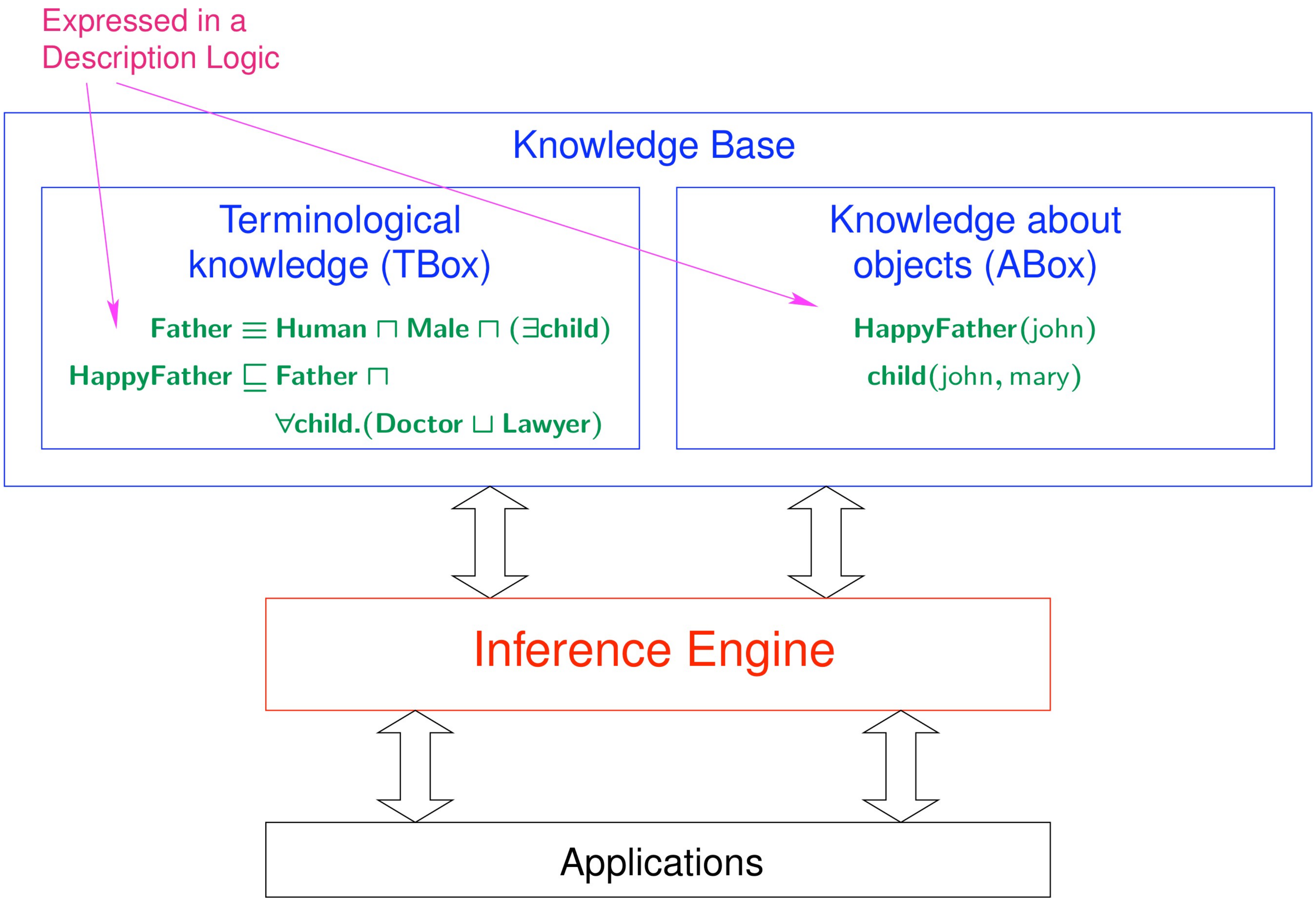
A mechanism to specify properties of objects (i.e., an ABox)

$\mathcal{A} = \{\text{HappyFather}(\text{john}), \text{hasChild}(\text{john}, \text{mary})\}$

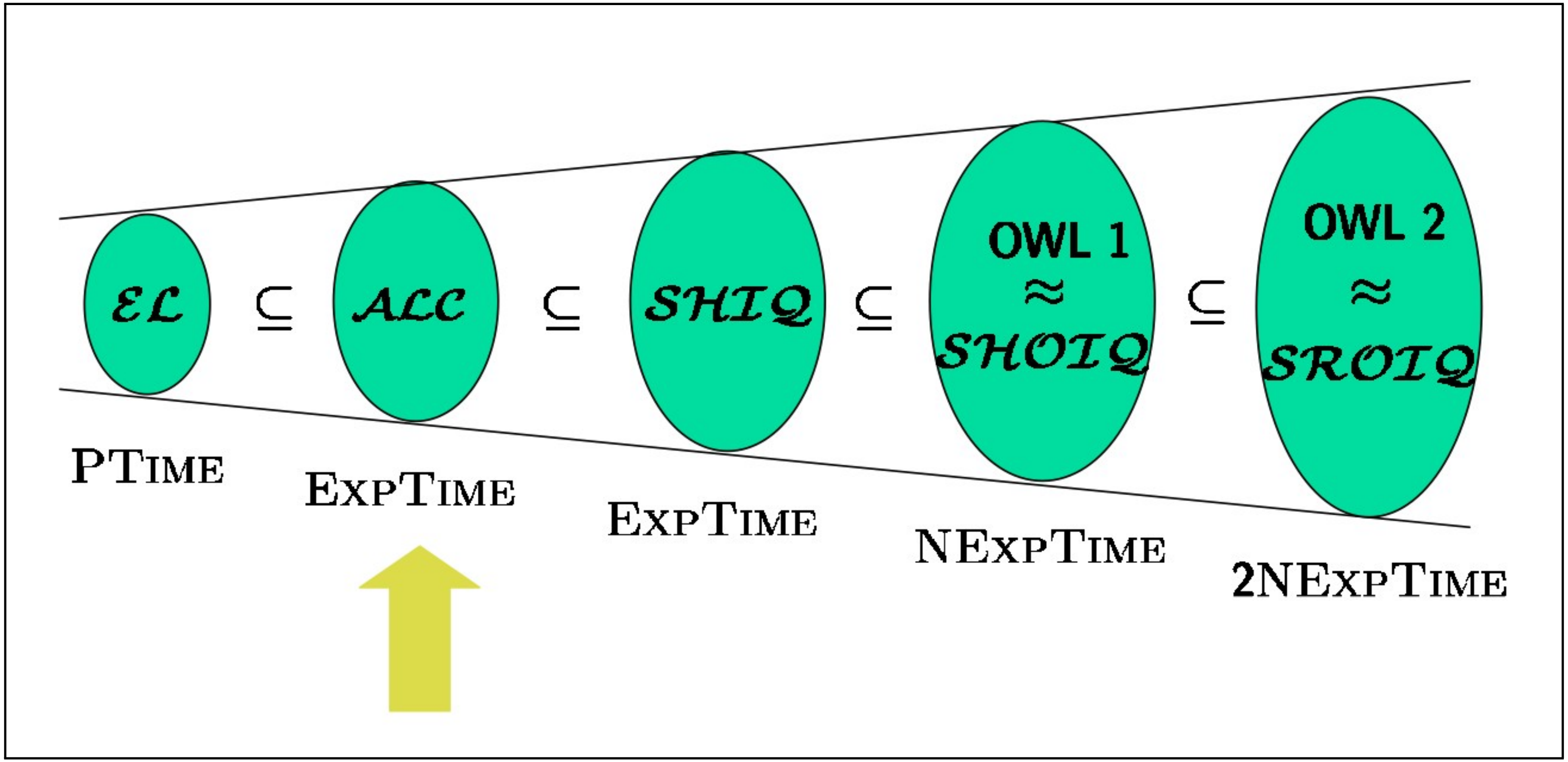
A set of inference services that allow to infer new properties on concepts, roles and objects, which are logical consequences of those explicitly asserted in the T-box and in the A-box

$$(\mathcal{T}, \mathcal{A}) \models \left\{ \begin{array}{l} \text{HappyFather} \sqsubseteq \exists \text{hasChild}.(\text{Doctor} \sqcup \text{Lawyer}) \\ \text{Doctor} \sqcup \text{Lawyer}(\text{mary}) \end{array} \right\}$$

Architecture of a Description Logic system



Many description logics



Syntax – ALC (AL with full concept negation)

- Formation rules:

  <Atomic> ::= A | B | … | P | Q | … | ⊥ | ⊤

  <wff> ::= <Atomic> | ¬ <wff> | <wff> ⊓ <wff> | <wff> ⊔ <wff> | ∀R.C | ∃R.C

  - ¬ (Mother ⊓ Father)

  "it cannot be both a mother and father"

  - Person ⊓ Female

    "persons that are female"

  - Person ⊓ ∃hasChild.⊤

  "(all those) persons that have a child"

  - Person ⊓ ∀hasChild.⊥

  "(all those) persons without a child"

  - Person ⊓ ∀hasChild.Female

  "persons all of whose children are female"

Syntax – ClassL as DL-language

- Introduction of the $\sqcup$ and elimination of roles $\forall$R.C and $\exists$R.C

- Formation rules:

  <Atomic> ::= A | B | ... | P | Q | ... | $\perp$ | $\top$

  <wff> ::= <Atomic> | ¬ <wff> | <wff> $\sqcap$ <wff> | <wff> $\sqcup$ <wff>

- The new language is a description language without roles which is ClassL (also called propositional DL)

NOTE: So far, we are considering DL without TBOX and ABox.

Syntax - *AL** Interpretation (Δ,I)

- $I(\bot) = \emptyset$ and $I(\top) = \Delta$ (full domain, "Universe")
- For every concept name A of L, $I(A) \subseteq \Delta$
- $I(\neg C) = \Delta \setminus I(C)$
- $I(C \sqcap D) = I(C) \cap I(D)$
- $I(C \sqcup D) = I(C) \cup I(D)$

The **SAME** as in **ClassL**

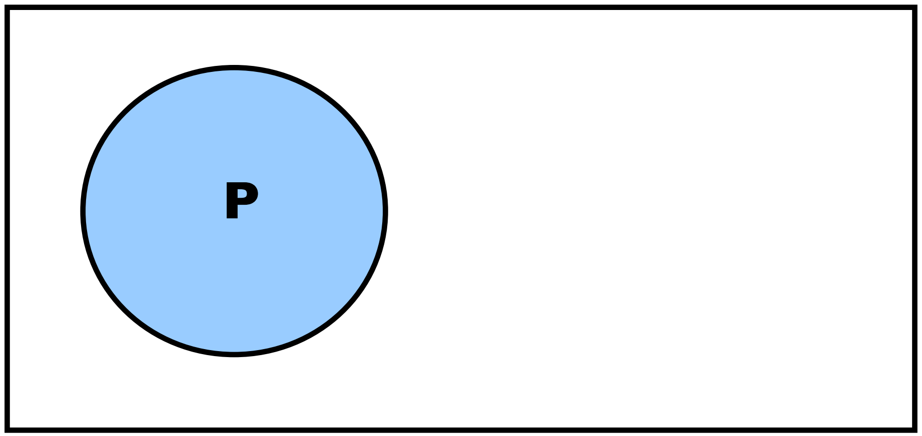
- For every role name R of L, $I(R) \subseteq \Delta \times \Delta$
- $I(\forall R.C) = \{a \in \Delta \mid \text{for all } b, \text{if } (a,b) \in I(R) \text{ then } b \in I(C)\}$
- $I(\exists R.\top) = \{a \in \Delta \mid \text{exists } b \text{ s.t. } (a,b) \in I(R)\}$
- $I(\exists R.C) = \{a \in \Delta \mid \text{exists } b \text{ s.t. } (a,b) \in I(R), b \in I(C)\}$
- $I(\geq nR) = \{a \in \Delta \mid |\{b \mid (a, b) \in I(R)\}| \geq n\}$
- $I(\leq nR) = \{a \in \Delta \mid |\{b \mid (a, b) \in I(R)\}| \leq n\}$

Semantics - Venn Diagrams and Class-Values

- By regarding propositions as classes, it is very convenient to use Venn diagrams
- Venn diagrams are used to represent extensional semantics of propositions in analogy of how truth-tables are used to represent intentional semantics
- Venn diagrams allow to compute a class valuation σ's value in polynomial time
- In Venn diagrams we use intersecting circles to represent the extension of a proposition, in particular of each atomic proposition
- The key idea is to use Venn diagrams to symbolize the extension of a proposition P by the device of shading the region corresponding to the proposition, as to indicate that P has a meaning (i.e., the extension of P is *not* empty).

Semantics - Venn Diagram of P, $\perp$

$\sigma(P)$



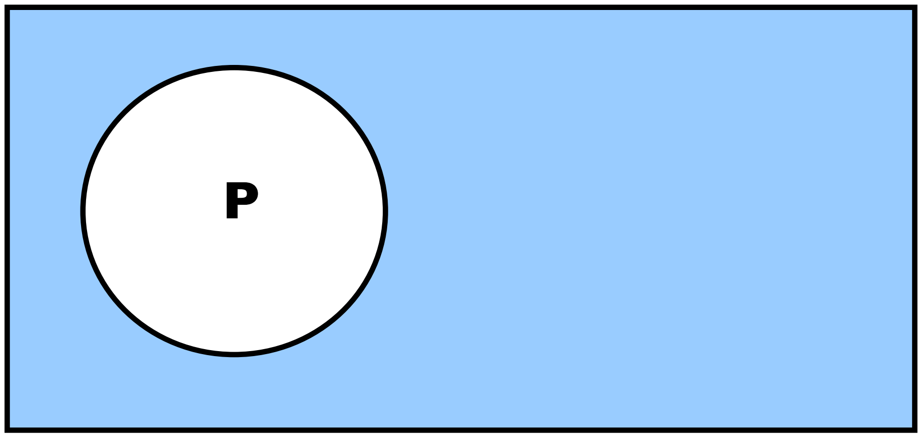
Venn diagrams are built starting from a "main box" which is used to represent the Universe U.

$\sigma(\perp)$

The falsehood symbol corresponds to the empty set.

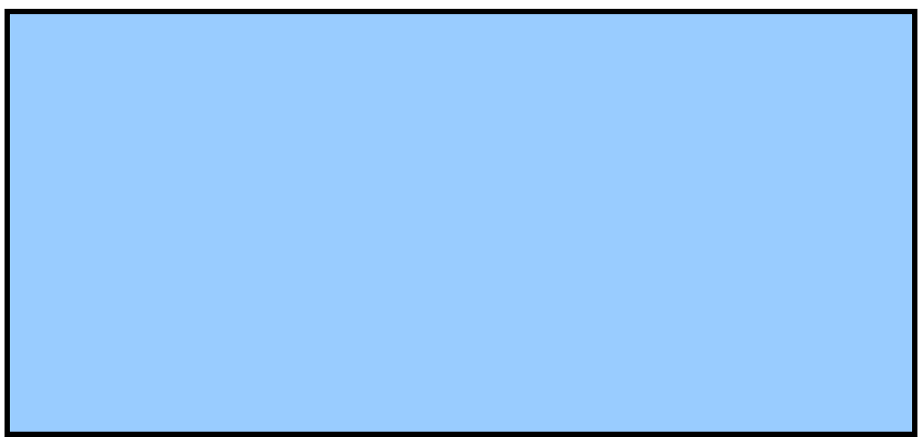
Semantics - Venn Diagram of ¬P, ⊤

σ(¬P)



¬P corresponds to the complement of P w.r.t. the universe U.

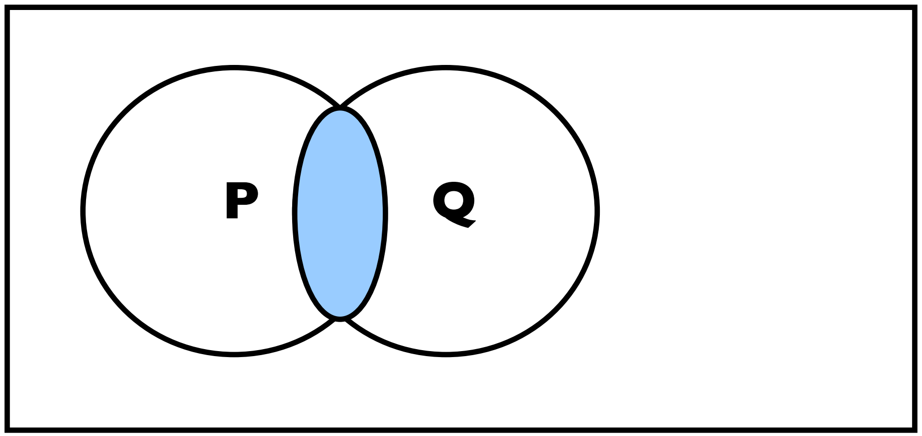
σ(⊤)



The truth symbol corresponds to the universe U.

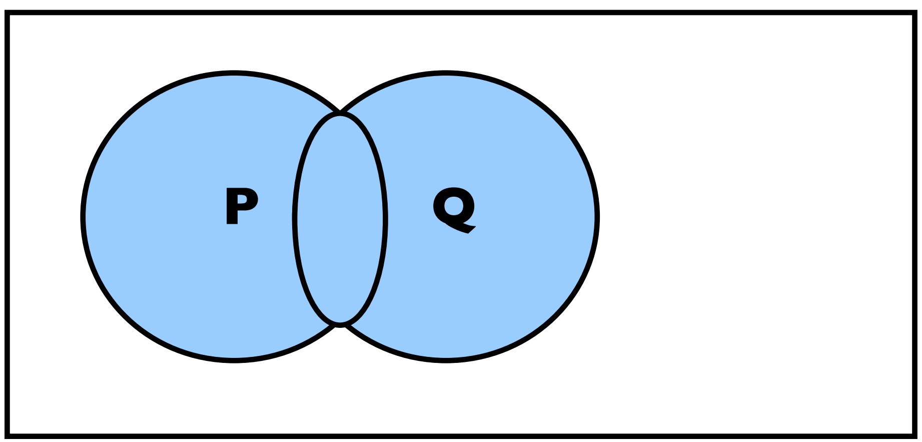
Semantics - Venn Diagram of P ⊓ Q and P ⊔ Q

$\sigma(P \sqcap Q)$



The intersection of P and Q

$\sigma(P \sqcup Q)$

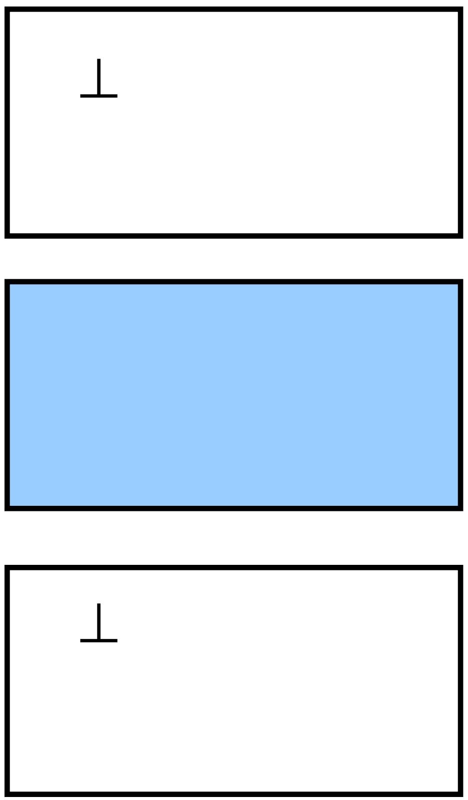


The union of P and Q

How to use Venn diagrams - exercise 1

- Prove by Venn diagrams that $\sigma(P) = \sigma(\neg\neg P)$

- Case $\sigma(P) = \emptyset$



$\sigma(P)$

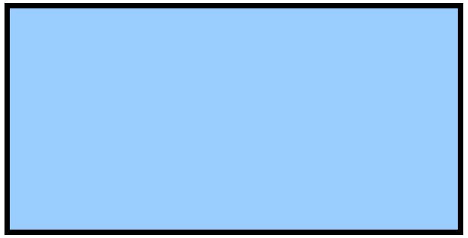
$\sigma(\neg P)$

$\sigma(\neg\neg P)$

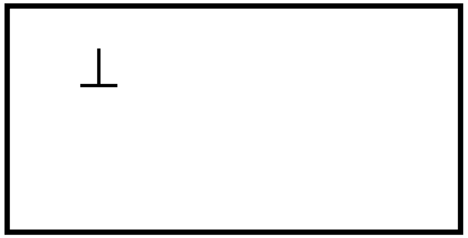
How to use Venn diagrams - exercise 1

- Prove by Venn diagrams that σ(P) = σ(¬¬P)

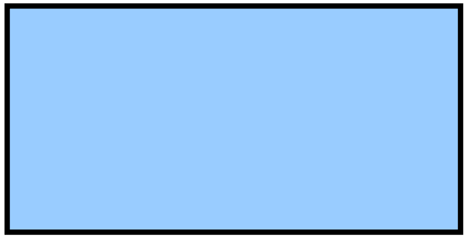
- Case σ(P) = U



σ(P)



σ(¬P)



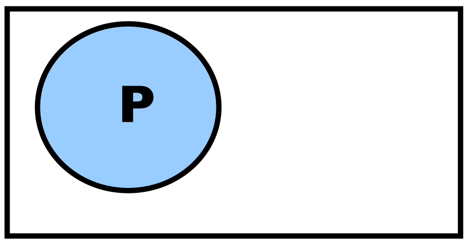
σ(¬¬P)

25

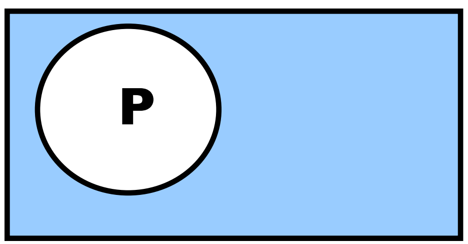
How to use Venn diagrams - exercise 1

- Prove by Venn diagrams that $\sigma(P) = \sigma(\neg\neg P)$

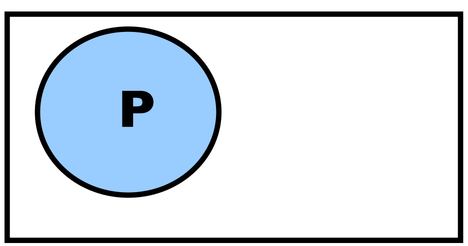
- Case $\sigma(P)$ not empty and different from U



$\sigma(P)$



$\sigma(\neg P)$

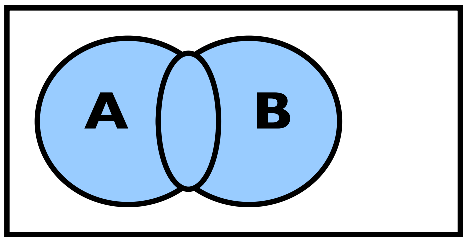


$\sigma(\neg\neg P)$

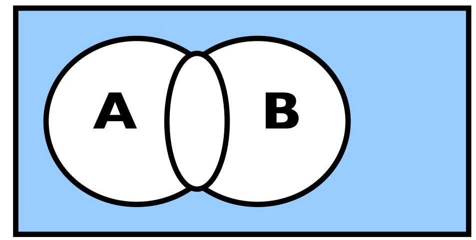
How to use Venn diagrams - exercise 2

- Prove by Venn diagrams that $\sigma(\neg(A \sqcup B)) = \sigma(\neg A \sqcap \neg B)$

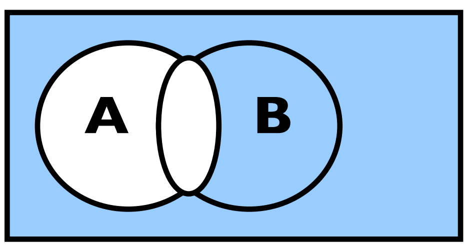
- Case $\sigma(A)$ and $\sigma(B)$ not empty (other cases as homework)



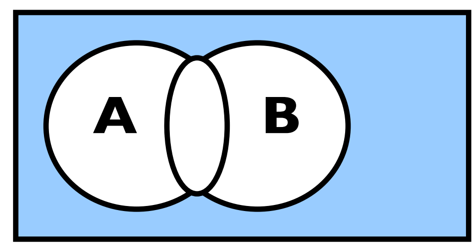
$\sigma(A \sqcup B)$



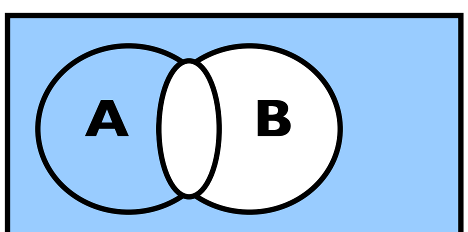
$\sigma(\neg(A \sqcup B))$



$\sigma(\neg A)$



$\sigma(\neg A \sqcap \neg B)$



$\sigma(\neg B)$

27

Semantics - Truth Relation (Satisfaction Relation)

- Let σ be a class-valuation on language L, we define the truth-relation (or class-satisfaction relation) $\models$ and write

$$\sigma \models P$$

(read: σ satisfies P) iff $\sigma(P) \neq \varnothing$

- Given a set of propositions Γ, we define

$$\sigma \models \Gamma$$

iff $\sigma \models \vartheta$ *for all formulas* $\vartheta \in \Gamma$

Semantics - Model and Satisfiability

- Let σ be a class valuation on language L. σ is a model of a proposition P (set of propositions Γ) iff σ satisfies P (Γ).

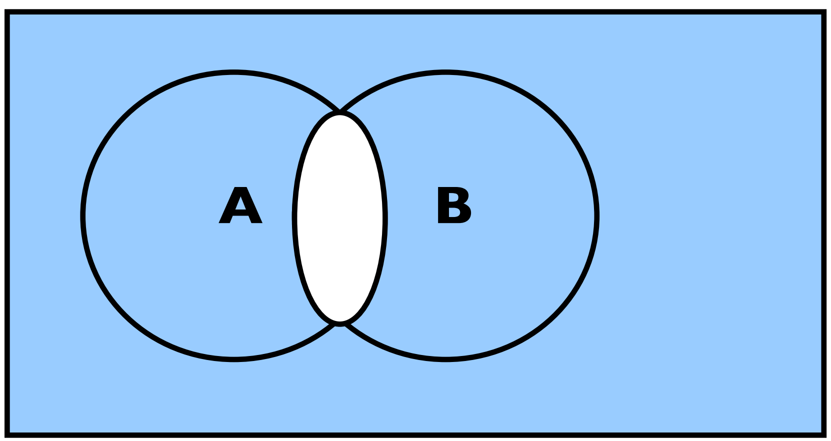
- P (Γ) is class-satisfiable if there is a class valuation σ such that $\sigma \vDash P$ ($\sigma \vDash \Gamma$).

Semantics - Satisfiability, an example

- Is the formula $P = \neg(A \sqcap B)$ satisfiable?

In other words, there exist a σ that satisfies P? YES!

In order to prove it we use Venn diagrams and it is enough to find <u>one</u>.



σ is <u>a</u> model for P

Semantics - Truth, satisfiability and validity

- Let σ be a class valuation on language L.
- P is true under σ if P is satisfiable ($\sigma \models P$)
- P is valid if $\sigma \models P$ for all σ (notation: $\models P$)
- In this case, P is called a tautology (always true)
- NOTE: the notions of 'true' and 'false' are relative to some truth valuation.

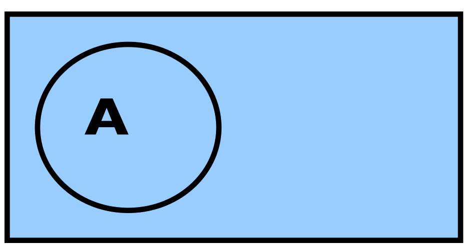
Semantics - Validity, an example

- Is the formula $P = A \sqcup \neg A$ valid?

  In other words, is P true for all σ? YES!

  In order to prove it we use Venn diagrams, but we need to discuss <u>all</u> cases.

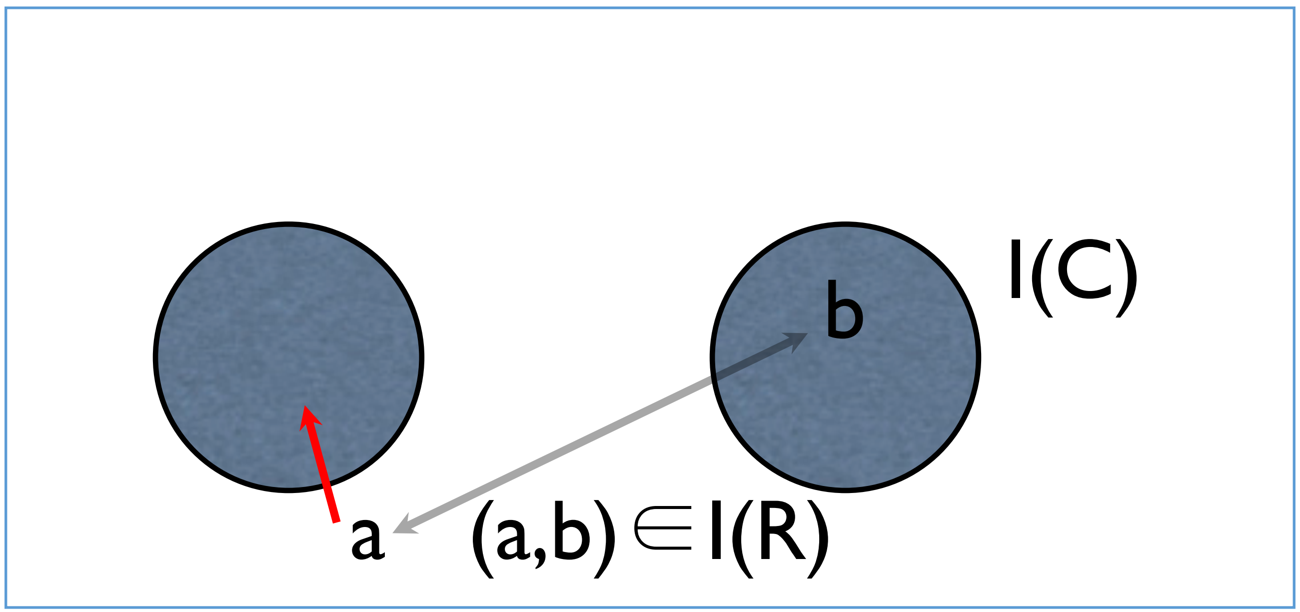
Case σ(A) empty:
if σ(A) is empty, then σ(¬A) is the universe U



Case σ(A) not empty:
if σ(A) is not empty, σ(¬A) covers all the other elements of U

Semantics - Interpretation of Existential Quantifier

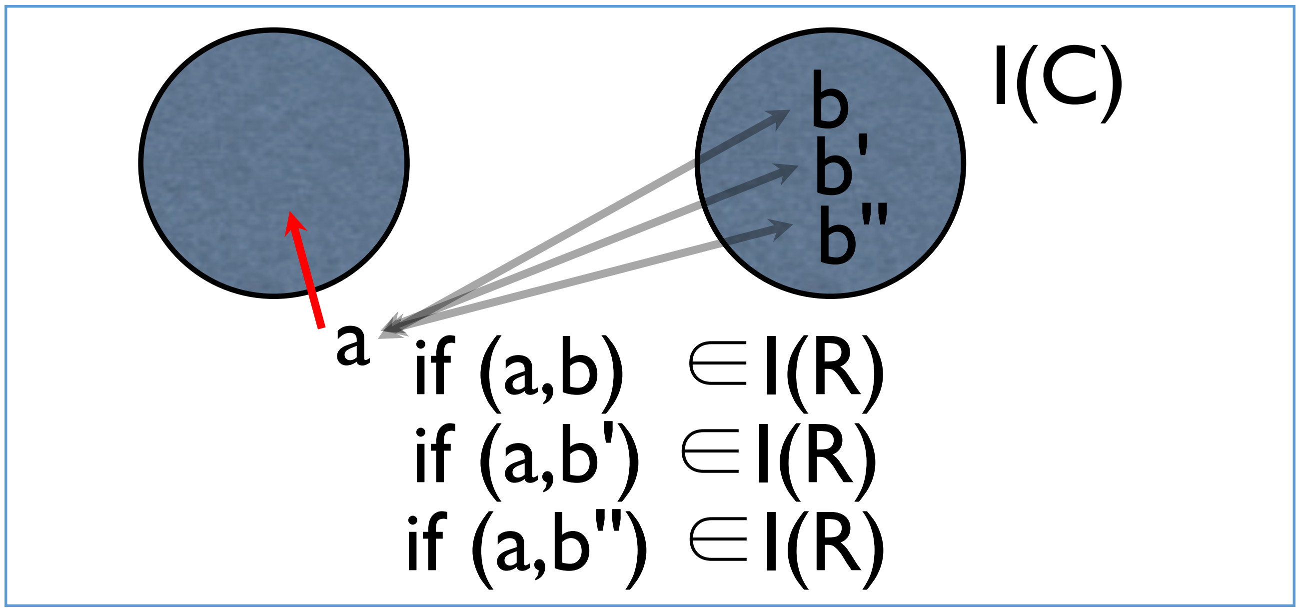
- $I(\exists R.C) = \{a \in \Delta \mid \text{exists } b \text{ s.t. } (a,b) \in I(R), b \in I(C)\}$



- Those a that have **some** value b in C with role R.

Semantics - Interpretation of Value Restriction

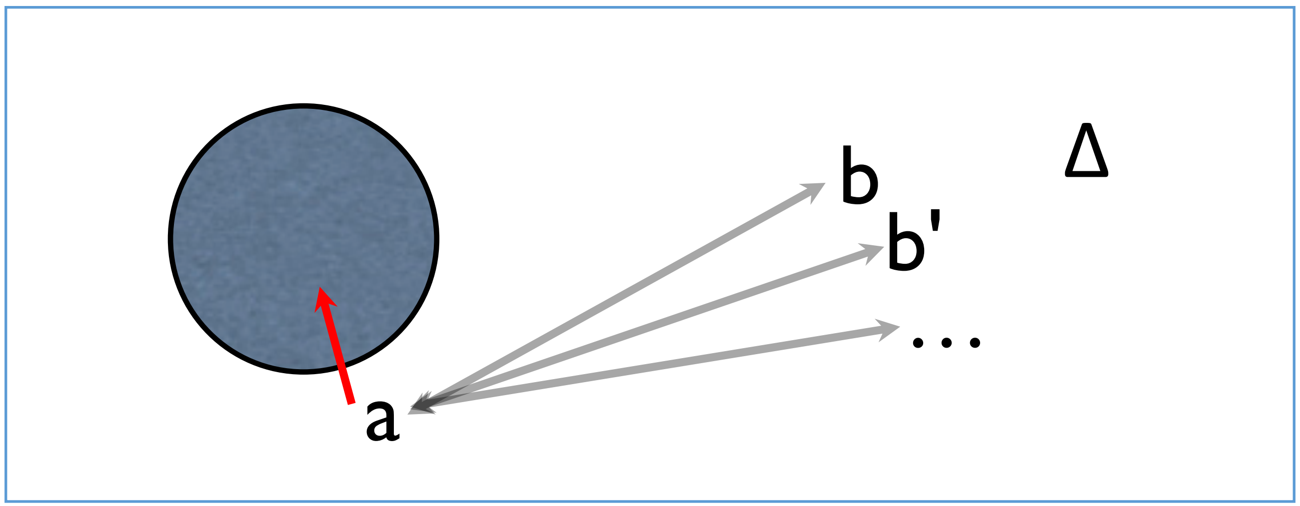
- $I(\forall R.C) = \{a \in \Delta \mid \text{for all } b, \text{ if } (a,b) \in I(R) \text{ then } b \in I(C)\}$



- Those a that have **only** values b in C with role R.

Semantics - Interpretation of Number Restriction

- $I(\geq nR) = \{a \in \Delta \mid |\{b \mid (a, b) \in I(R)\}| \geq n\}$



$$|\{b \mid (a, b) \in I(R)\}| \geq n$$

- Those a that have relation R to **at least n** individuals.

Reasoning on Class-Propositions

Given a class-propositions P we want to reason about the following:

❑ Model checking	Does σ satisfy P? ($\sigma \models P$?)
❑ Satisfiability	Is there any σ such that $\sigma \models P$?
❑ Unsatisfiability	Is it true that there are no σ satisfying P?
❑ Validity	Is P a tautology? (true for all σ)