Mathematical Logics
Description Logic: Introduction

Fausto Giunchiglia and Mattia Fumagalli

University of Trento

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

World → Mental Model

Language L expresses Theory T

Domain D grounds Model M

SEMANTIC GAP

Causes

Represents
Logical Model

Language L expresses Theory T

Domain D represents Model M

World

SEMANTIC GAP

EXPLANATION

 causes

represents

interpretation

entailment

grounds
Logical Model

\[ L = \text{"Monkey, Above, Banana, Near, Get, monkey#1, banana#2, tree, } \cap, \cup, \neg, \exists, \forall \text{ ..."} \]

\[ \text{TBOX} = \text{"MonkeyGetBanana } \equiv \text{ Monkey } \cap \forall \text{ Get.Banana"} \]

\[ D: \{ \text{monkey#1, banana#2} \} \]

\[ \text{ABOX: } \text{"MonkeyGetBanana(monkey#1), Banana(banana#2)"} \]

\[ T, A \not\vdash \forall \text{ Get.Banana} \]
The syntax of ClassL is similar to PL

Alphabet of symbols $\Sigma_0$

- **Descriptive**
  - Constants
    - one proposition only
    - $A, B, C \ldots$
  - Variables
    - they can be substituted by any proposition or formula
    - $P, Q, \psi \ldots$

- **Logical**
  - $\land, \lor, \neg$

NOTE: not only characters but also words (composed by several characters) like “monkey” are descriptive symbols
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
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
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
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).
A **DL** is characterized by:

A **description language**: how to form concepts and roles

\[ \text{Human} \sqcap \text{Male} \sqcap \exists \text{hasChild}.T \sqcap \forall \text{hasChild}.(\text{Doctor} \sqcup \text{Lawyer}) \]

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

\[
\begin{aligned}
\text{Father} &\equiv \text{Human} \sqcap \text{Male} \sqcap \exists \text{hasChild}.T \\
T &\equiv \text{HappyFather} \sqsubseteq \text{Father} \sqcap \forall \text{hasChild}.(\text{Doctor} \sqcup \text{Lawyer}) \\
\text{hasFather} &\equiv \exists \text{hasParent}
\end{aligned}
\]

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

\[ A = \{\text{HappyFather (john)}, \text{hasChild (john, mary )}\} \]

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

\[ (T, A) \models \begin{cases}
\text{HappyFather} \sqsubseteq \exists \text{hasChild}.(\text{Doctor} \sqcup \text{Lawyer}) \\
\text{Doctor} \sqcup \text{Lawyer} (\text{mary})
\end{cases} \]
Architecture of a Description Logic system

Knowledge Base

Terminological knowledge (TBox)

\[ \text{Father} \equiv \text{Human} \sqcap \text{Male} \sqcap (\exists \text{child}) \]
\[ \text{HappyFather} \sqsubseteq \text{Father} \sqcap \forall \text{child}. (\text{Doctor} \sqcap \text{Lawyer}) \]

Knowledge about objects (ABox)

\[ \text{HappyFather}(\text{john}) \]
\[ \text{child}(\text{john}, \text{mary}) \]

Inference Engine

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

Formation rules:

- $<\text{Atomic}> ::= A \mid B \mid \ldots \mid P \mid Q \mid \ldots \mid \bot \mid \top$
- $<\text{wff}> ::= <\text{Atomic}> \mid \neg <\text{wff}> \mid <\text{wff}> \sqcap <\text{wff}> \mid <\text{wff}> \sqcup <\text{wff}>$

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 ($\Delta, I$)

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

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

The SAME as in Class$L$
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 $\sigma$’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).
Venn diagrams are built starting from a “main box” which is used to represent the Universe U.

The falsehood symbol corresponds to the empty set.
¬P corresponds to the complement of P w.r.t. the universe U.

The truth symbol corresponds to the universe U.
Semantics - Venn Diagram of $P \cap Q$ and $P \cup Q$

\[ \sigma(P \cap Q) \]

The intersection of $P$ and $Q$

\[ \sigma(P \cup Q) \]

The union of $P$ and $Q$
Prove by Venn diagrams that $\sigma(P) = \sigma(\neg\neg P)$

Case $\sigma(P) = \emptyset$

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

Case $\sigma(P) = U$

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

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

\begin{itemize}
  \item $\sigma(P)$
  \item $\sigma(\neg P)$
  \item $\sigma(\neg \neg P)$
\end{itemize}
Prove by Venn diagrams that $\sigma(\neg(A \sqcup B)) = \sigma(\neg A \cap \neg B)$

Case $\sigma(A)$ and $\sigma(B)$ not empty (other cases as homework)
Let $\sigma$ be a class-valuation on language $L$, we define the truth-relation (or class-satisfaction relation) $\models$ and write

$$\sigma \models P$$

(read: $\sigma$ satisfies $P$) iff $\sigma(P) \neq \emptyset$

Given a set of propositions $\Gamma$, we define

$$\sigma \models \Gamma$$

iff $\sigma \models \varnothing$ for all formulas $\varnothing \in \Gamma$
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 σ ⊨ P (σ ⊨ Γ).
Is the formula $P = \neg(A \cap B)$ satisfiable?

In other words, there exist a $\sigma$ that satisfies $P$?  

YES!

In order to prove it we use Venn diagrams and it is enough to find one.

$\sigma$ is a model for $P$
Let $\sigma$ be a class valuation on language $L$.

- P is **true under** $\sigma$ if $P$ is satisfiable ($\sigma \models P$)

- P is **valid** if $\sigma \models P$ for all $\sigma$ (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.
Is the formula $P = A \uplus \neg A$ valid?
In other words, is $P$ true for all $\sigma$? YES!

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

Case $\sigma(A)$ empty:
if $\sigma(A)$ is empty, then $\sigma(\neg A)$ is the universe $U$

Case $\sigma(A)$ not empty:
if $\sigma(A)$ is not empty, $\sigma(\neg A)$ 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.
\( I(\forall R.C) = \{ a \in \Delta \mid \text{for all } b, \text{ if } (a,b) \in I(R) \text{ then } b \in I(C) \} \)

Those \( a \) that have \textbf{only} values \( b \) in \( C \) with role \( R \).
Semantics - Interpretation of Number Restriction

\[ \{a \in \Delta \mid |\{b \mid (a, b) \in I(R)\}| \geq n\} \]

Those \( a \) that have relation \( R \) to \textbf{at least} \( n \) individuals.
Given a class-propositions $P$ we want to reason about the following:

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