Automated Reasoning in Artificial Intelligence:

# Introduction to Description Logic

**Szymon Klarman**

(*part of the content based on the tutorial by* **Stefan Schlobach**)

szymon.klarman@gmail.com

VU University Amsterdam, 2009-2012

# Overview of the module

**Lectures**:

(I) Modeling concepts in Description Logics

(II) Ontologies and reasoning tasks (*laptops needed*)

(III) Tableau algorithm for Description Logics

**Assignment**:

Implement a Description Logic reasoner using the LoTREC toolkit.

(IV) LoTREC tutorial (*laptops needed*)

**Tools**:

- Protégé (`http://protege.stanford.edu/`)
- LoTREC (`http://www.irit.fr/Lotrec/`)

# Plan for today

- Knowledge Representation and Description Logics (DLs)
- Syntax and semantics of concepts in the language $\mathcal{ALC}$
- Other DL languages
- Design philosophy and research problems

# KR and Description Logics

*Knowledge Representation focuses on the study of methods for building high-level descriptions of the world to support design of intelligent systems.*

Why do we want to do KR? Because:

- it is better to separate programming from knowledge models,
- one can use generic, domain-independent problem solvers.

*Description Logics are a family of (concept-based) knowledge representation formalisms that represent the knowledge about an application domain in terms of a terminology of concepts and a description of the properties of objects that exist in the domain.*

F. Baader, and W. Nutt, *Description Logics Handbook*

# Basic intuition

*I know the meaning of some astronomical concepts*:

1. A planet is a celestial body that orbits around some star.
2. Moons orbit only around planets.
3. Planets and stars are disjoint classes of objects.

*I also know some facts*:

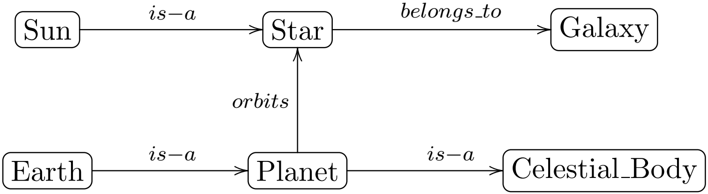
1. Earth is a planet.
2. The Moon orbits around the Earth.

*Could I tell it all to my computer and get the following inferences*?

1. The Moon is a moon.
2. The Moon cannot orbit around any star.
3. Moons and planets are disjoint classes of objects.

# Origins: cognitive inspirations

- *Semantic Networks* (1967) for representing contents of dictionaries.
- Knowledge represented via labeled graphs and reasoning based on graph operations.




- a *user-friendly interface*,
- no formal semantics (object vs. concept nodes, what is *is-a*?),
- expressive and reasoning capabilities not clear.

Therefore:

- it is impossible to design robust reasoners,
- different systems might deliver different inferences.

# Origins: logical inspirations

Why use logic as the basis for KR? Because:

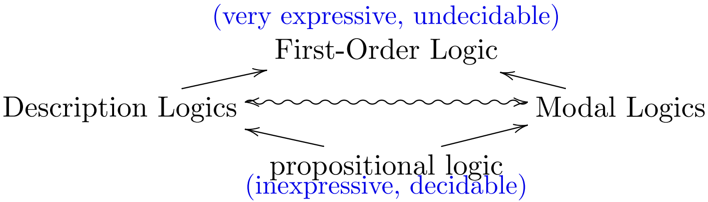
- logical languages have precisely defined *syntax* and *semantics*,
- reasoning can be based on *logical entailment* and supported by means of automated theorem proving techniques,
- many problems can be much better understood when rendered in logic (e.g. consistency, complexity of reasoning).

But which logic?

- logical syntaxes appear usually heavy and unattractive,
- first attempts of formalizing semantics based on First-Order Logic (1979).

# Description Logics

- Provide a user-friendly, concept-oriented syntax, maintaining formal semantics.
- Offer features especially useful from the KR perspective.
- Remain expressive but decidable:




- Also known as: *terminological systems*, *concept languages*,
- Pre-DL systems (mid-80's); early DL systems (early 90's); the mature form and popularity boom since late 90's.

# $\mathcal{ALC}$: Syntax

$\mathcal{ALC}$= Attributive Language with Complement

The *vocabulary* of a Description Logic language includes:

- *concept names*, e.g. $Man, Parent, Car$ $(A, B, C \ldots)$,
- *role names*, e.g. $biggerThan, likes, locatedIn$ $(r, s \ldots)$.

Complex *concept descriptions* are built from atomic terms by means of the *constructors*:

| | | | | |
|---|---|---|---|---|
| $C, D$ | $\rightarrow$ | $A$ | atomic concept | |
| | | $\top$ | universal concept | "*thing*" |
| | | $\bot$ | bottom concept | "*nothing*" |
| | | $\neg C$ | complement | "*not*" |
| | | $C \sqcap D$ | intersection | "*and*" |
| | | $C \sqcup D$ | union | "*or*" |
| | | $\exists r.C$ | existential restriction | "*some*" |
| | | $\forall r.C$ | universal restriction | "*only*" |

# Exercise: modeling $\mathcal{ALC}$ concepts

*"Any artwork is created by an artist. A sculpture is an artwork. A painting is an artwork that is not a sculpture. A painter is someone who painted a painting. A sculptor is someone who sculptured an artwork and only create sculptures. If an artwork is created by an artist, he has either painted or sculptured it."*

- Determine the set of atomic concepts and roles.

# Exercise: modeling $\mathcal{ALC}$ concepts

*"Any artwork is created by an artist. A sculpture is an artwork. A painting is an artwork that is not a sculpture. A painter is someone who painted a painting. A sculptor is someone who sculptured an artwork and only create sculptures. If an artwork is created by an artist, he has either painted or sculptured it."*

- Determine the set of atomic concepts and roles.
- Solution:
  - Atomic concepts = $\{Artwork, Artist, Sculptor, Painter, Painting, Sculpture\}$
  - Atomic roles = $\{created, created\_by, painted, sculptured\}$

# Exercise: modeling $\mathcal{ALC}$ concepts

- Model the following complex concepts:
  - a piece of art that is not a sculpture
  - someone, who painted a painting
  - someone, who sculptured a piece of art, and only created sculptures

# Exercise: modeling $\mathcal{ALC}$ concepts

- Model the following complex concepts:
  - a piece of art that is not a sculpture
  - someone, who painted a painting
  - someone, who sculptured a piece of art, and only created sculptures
- Solution:
  - $Artwork \sqcap \neg Sculpture$

# Exercise: modeling $\mathcal{ALC}$ concepts

- Model the following complex concepts:
  - a piece of art that is not a sculpture
  - someone, who painted a painting
  - someone, who sculptured a piece of art, and only created sculptures
- Solution:
  - $Artwork \sqcap \neg Sculpture$
  - $\exists painted.Painting$

# Exercise: modeling $\mathcal{ALC}$ concepts

- Model the following complex concepts:
  - a piece of art that is not a sculpture
  - someone, who painted a painting
  - someone, who sculptured a piece of art, and only created sculptures
- Solution:
  - $Artwork \sqcap \neg Sculpture$
  - $\exists painted.Painting$
  - $\exists sculptured.Artwork \sqcap \forall created.Sculpture$

# $\mathcal{ALC}$: Semantics

The semantics is given through *interpretations*. An interpretation is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty *domain of individuals* and $\cdot^\mathcal{I}$ is an *interpretation function*, which maps:

- $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, i.e. concept names to subsets of $\Delta^\mathcal{I}$,
- $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, i.e. role names to subsets of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

$\cdot^\mathcal{I}$ is inductively extended over complex concept descriptions:

$$
\begin{aligned}
\top^\mathcal{I} &= \Delta^\mathcal{I} \\
\bot^\mathcal{I} &= \emptyset \\
(\neg C)^\mathcal{I} &= \Delta^\mathcal{I} \backslash C^\mathcal{I} \\
(C \sqcap D)^\mathcal{I} &= C^\mathcal{I} \cap D^\mathcal{I} \\
(C \sqcup D)^\mathcal{I} &= C^\mathcal{I} \cup D^\mathcal{I} \\
(\exists r.C)^\mathcal{I} &= \{x \in \Delta^\mathcal{I} \mid \exists y.(x, y) \in r^\mathcal{I} \wedge y \in C^\mathcal{I}\} \\
(\forall r.C)^\mathcal{I} &= \{x \in \Delta^\mathcal{I} \mid \forall y.(x, y) \in r^\mathcal{I} \rightarrow y \in C^\mathcal{I}\}
\end{aligned}
$$

# Exercise: semantics of $\mathcal{ALC}$ concepts

- Assume the following base interpretation:
  $\Delta^{\mathcal{I}} = \{rembrandt, michelangelo, rodin, nightwatch, david, sixtChappel, thinker\}$

$Artwork^{\mathcal{I}} = \{nightwatch, sixtChappel, thinker, david\}$,
$Artist^{\mathcal{I}} = \{rembrandt, rodin, michelangelo\}$
$Sculptor^{\mathcal{I}} = \{rodin, michelangelo\}$ $Sculpture^{\mathcal{I}} = \{thinker, david\}$
$Painter^{\mathcal{I}} = \{rembrandt, michelangelo\}$
$Painting^{\mathcal{I}} = \{nightwatch, sixtChappel\}$
$painted^{\mathcal{I}} = \{(rembrandt, nightwatch), (michelangelo, sixtChappel),$
$sculptured^{\mathcal{I}} = \{(rodin, thinker), (michelangelo, david)\}$
$created^{\mathcal{I}} = \{(rembrandt, nightwatch), (michelangelo, sixtChappel), (michelangelo, david), (rodin, thinker)\}$

# Exercise: semantics of $\mathcal{ALC}$ concepts

- Compute the semantics of the following concepts:
  1. $Artwork \sqcap \neg Sculpture$
  2. $\exists painted.Painting$
  3. $\exists sculptured.Artwork \sqcap \forall created.Sculpture$
  4. $\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture)$
  5. $\forall created.Painting \sqcap \exists created.\top$
  6. $\exists created.Painting$

# Exercise: semantics of $\mathcal{ALC}$ concepts

- Compute the semantics of the following concepts:
  1. $Artwork \sqcap \neg Sculpture$
  2. $\exists painted.Painting$
  3. $\exists sculptured.Artwork \sqcap \forall created.Sculpture$
  4. $\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture)$
  5. $\forall created.Painting \sqcap \exists created.\top$
  6. $\exists created.Painting$
- Solution:
  1. $(Artwork \sqcap \neg Sculpture)^{\mathcal{I}} = \{nightwatch, sixtChappel\}$

# Exercise: semantics of $\mathcal{ALC}$ concepts

- Compute the semantics of the following concepts:
  1. $Artwork \sqcap \neg Sculpture$
  2. $\exists painted.Painting$
  3. $\exists sculptured.Artwork \sqcap \forall created.Sculpture$
  4. $\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture)$
  5. $\forall created.Painting \sqcap \exists created.\top$
  6. $\exists created.Painting$
- Solution:
  1. $(Artwork \sqcap \neg Sculpture)^{\mathcal{I}} = \{nightwatch, sixtChappel\}$
  2. $(\exists painted.Painting)^{\mathcal{I}} = \{rembrandt, michelangelo\}$

# Exercise: semantics of $\mathcal{ALC}$ concepts

- Compute the semantics of the following concepts:
  1. $Artwork \sqcap \neg Sculpture$
  2. $\exists painted.Painting$
  3. $\exists sculptured.Artwork \sqcap \forall created.Sculpture$
  4. $\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture)$
  5. $\forall created.Painting \sqcap \exists created.\top$
  6. $\exists created.Painting$
- Solution:
  1. $(Artwork \sqcap \neg Sculpture)^{\mathcal{I}} = \{nightwatch, sixtChappel\}$
  2. $(\exists painted.Painting)^{\mathcal{I}} = \{rembrandt, michelangelo\}$
  3. $(\exists sculptured.Artwork \sqcap \forall created.Sculpture)^{\mathcal{I}} = \{rodin\}$

# Exercise: semantics of $\mathcal{ALC}$ concepts

- Compute the semantics of the following concepts:
  1. $Artwork \sqcap \neg Sculpture$
  2. $\exists painted.Painting$
  3. $\exists sculptured.Artwork \sqcap \forall created.Sculpture$
  4. $\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture)$
  5. $\forall created.Painting \sqcap \exists created.\top$
  6. $\exists created.Painting$
- Solution:
  1. $(Artwork \sqcap \neg Sculpture)^{\mathcal{I}} = \{nightwatch, sixtChappel\}$
  2. $(\exists painted.Painting)^{\mathcal{I}} = \{rembrandt, michelangelo\}$
  3. $(\exists sculptured.Artwork \sqcap \forall created.Sculpture)^{\mathcal{I}} = \{rodin\}$
  4. $(\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture))^{\mathcal{I}} = \varnothing$

# Exercise: semantics of $\mathcal{ALC}$ concepts

- Compute the semantics of the following concepts:
  1. $Artwork \sqcap \neg Sculpture$
  2. $\exists painted.Painting$
  3. $\exists sculptured.Artwork \sqcap \forall created.Sculpture$
  4. $\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture)$
  5. $\forall created.Painting \sqcap \exists created.\top$
  6. $\exists created.Painting$
- Solution:
  1. $(Artwork \sqcap \neg Sculpture)^{\mathcal{I}} = \{nightwatch, sixtChappel\}$
  2. $(\exists painted.Painting)^{\mathcal{I}} = \{rembrandt, michelangelo\}$
  3. $(\exists sculptured.Artwork \sqcap \forall created.Sculpture)^{\mathcal{I}} = \{rodin\}$
  4. $(\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture))^{\mathcal{I}} = \varnothing$
  5. $(\forall created.Painting \sqcap \exists created.\top)^{\mathcal{I}} = \{rembrandt\}$

# Exercise: semantics of $\mathcal{ALC}$ concepts

- Compute the semantics of the following concepts:
  1. $Artwork \sqcap \neg Sculpture$
  2. $\exists painted.Painting$
  3. $\exists sculptured.Artwork \sqcap \forall created.Sculpture$
  4. $\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture)$
  5. $\forall created.Painting \sqcap \exists created.\top$
  6. $\exists created.Painting$
- Solution:
  1. $(Artwork \sqcap \neg Sculpture)^{\mathcal{I}} = \{nightwatch, sixtChappel\}$
  2. $(\exists painted.Painting)^{\mathcal{I}} = \{rembrandt, michelangelo\}$
  3. $(\exists sculptured.Artwork \sqcap \forall created.Sculpture)^{\mathcal{I}} = \{rodin\}$
  4. $(\forall created.Sculpture \sqcap \exists created.(Artwork \sqcap \neg Sculpture))^{\mathcal{I}} = \varnothing$
  5. $(\forall created.Painting \sqcap \exists created.\top)^{\mathcal{I}} = \{rembrandt\}$
  6. $(\exists created.Painting)^{\mathcal{I}} = \{rembrandt, michelangelo\}$

# Meaning-preserving concept transformations

Because of well-defined semantics we can see that certain expressions in different syntactic forms have the same meaning. For instance:

- $\neg\top = \bot$
  *Proof*: $(\neg\top)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \Delta^{\mathcal{I}} = \emptyset = \bot^{\mathcal{I}}$
- $\neg\bot = \top$
- $\neg\neg C = C$
  *Proof*: $(\neg\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus (\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}) = (\Delta^{\mathcal{I}} \setminus \Delta^{\mathcal{I}}) \cup C^{\mathcal{I}} = C^{\mathcal{I}}$
- $\neg(C \sqcap D) = \neg C \sqcup \neg D$
  *Proof*: $(\neg(C \sqcap D))^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus (C^{\mathcal{I}} \cap D^{\mathcal{I}}) = (\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}) \cup (\Delta^{\mathcal{I}} \setminus D^{\mathcal{I}}) = (\neg C \sqcup \neg D)^{\mathcal{I}}$
- $\neg(C \sqcup D) = \neg C \sqcap \neg D$
- $\neg\forall r.C = \exists r.\neg C$
  *Proof*: $(\neg\forall r.C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus \{x \in \Delta^{\mathcal{I}} \mid \forall y.(x,y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\} =$
  $= \{x \in \Delta^{\mathcal{I}} \mid \neg(\forall y.(x,y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.(x,y) \in r^{\mathcal{I}} \wedge y \notin C^{\mathcal{I}}\} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.(x,y) \in r^{\mathcal{I}} \wedge y \in (\neg C)^{\mathcal{I}}\} = (\exists r.\neg C)^{\mathcal{I}}$
- $\neg\exists r.C = \forall r.\neg C$

# Other DL constructors

There are many other available constructors:

- *atomic complement*: $\neg A$
- *limited existential restriction*: $\exists r.\top$
- *nominal*: $\{a\}$
- *number restrictions*: $\leq n\ r,\ \geq n\ r, \leq n\ r.C,\ \geq n\ r.C$
- *role compositions*: $r \circ s$
- *role properties*: inverse, symmetric, transitive, reflexive, etc.
- *datatypes*: numbers, strings, etc.

and more....

For example:

$$Course \sqcap \exists taughtBy.(\{frank\} \sqcup \{annette\})$$
$$Mother \sqcap \leq 2\ hasChild.Male \sqcap \geq 3\ hasChild.Female$$
$$TVShow \sqcap \exists watches^{-}.(Spectator \sqcap \forall watches.Comedy)$$
$$Event \sqcap \exists hasTime.\text{“}\texttt{2002-05-30T09:00:00}\text{”}$$

# DL languages

There is a traditional code for naming particular DL building blocks:

| $\neg A$ | $C \sqcap D$ | $\forall r.C$ | $\exists r.\top$ | $\neg C$ | $C \sqcup D$ | $\exists r.C$ | $\{a\}$ | $r^-$ |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{AL}$ | $\mathcal{AL}$ | $\mathcal{AL}$ | $\mathcal{AL}$ | $\mathcal{C}$ | $\mathcal{U}$ | $\mathcal{E}$ | $\mathcal{O}$ | $\mathcal{I}$ |

You can add (or remove) features from $\mathcal{AL}$ ($\mathcal{A}$ttributive $\mathcal{L}$anguage) to obtain more (or less) expressive DLs. For instance:

- $\mathcal{ALC} = \mathcal{AL} + \mathcal{C} = \mathcal{AL} + \mathcal{U} + \mathcal{E}$
- $\mathcal{EL} = \mathcal{AL} - (\forall r.C) - (\neg A) + \mathcal{E}$
- $\mathcal{SROIQ}(D)$ = all above and more

# Expressiveness vs. complexity

There is a trade-off between *expressiveness* of a language and the *complexity of reasoning* in it:

| DL | complexity |
|---|---|
| $\mathcal{EL}$ | PTIME |
| $\mathcal{ALC}$ | EXPTIME-complete |
| $\vdots$ | $\vdots$ |
| $\mathcal{SROIQ}(D)$ | N2EXPTIME-complete |

DL Complexity Navigator: `http://www.cs.man.ac.uk/~ezolin/dl/`

Different properties facilitate different *applications*:

- $\mathcal{EL}$: large but simple terminologies, e.g. SNOMED
- $\mathcal{SROIQ}(D)$: Web Ontology Language OWL 2 DL
- $\mathcal{ALC}$: good for research and teaching DLs ;)

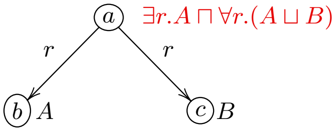
# Relationships to other logics

The relationships of DLs to other logics are quite well understood.

| DL | FOL | Modal Logic | Propositional Logic |
|---|---|---|---|
| $A$ | $A(x)$ | $p_A$ | $p_A$ |
| $r$ | $r(x,y)$ | access. relation $r$ | inexpressible |
| $\exists r.A$ | $\exists y.(r(x,y) \wedge A(y))$ | $\Diamond_r p_A$ | inexpressible |

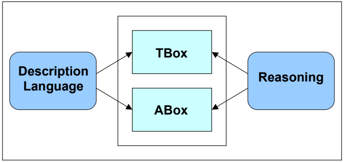
In particular, concepts of $\mathcal{ALC}$ are *notational variants* of modal logic formulas in $\mathbf{K}_n$. DL interpretations can be seen as *Kripke models*.

$$\Delta^{\mathcal{I}} = \{a, b, c\}$$
$$r^{\mathcal{I}} = \{(a,b), (a,c)\}$$
$$A^{\mathcal{I}} = \{b\}$$
$$B^{\mathcal{I}} = \{c\}$$



# Philosophy of Description Logics

- Separate terminological part of knowledge (relations between concepts) from the assertional part (descriptions of objects).




- Allow incomplete knowledge: The Open World Assumption.
- While developing, keep balance between theory and practice.
- Stay modular — find DLs with interesting compositions of constructors and for each one:
  - understand its properties (expressiveness, complexity),
  - develop well-behaved reasoning tools.

# Research on DLs

Research on DLs has lead to *important results* in KR, e.g.:

- expressivity-complexity trade-off,
- extensions to tableau-based techniques + optimizations e.g., FaCT (1998), Racer, Pellet.

*Application domains* include: (software) engineering, *e*-Science, bioinformatics (SNOMED CT >300k clinical terms), Semantic Web (foundation for Web Ontology Languages), and many others.

*Current research* focuses on:

- coupling DLs with database technologies,
- efficient query answering,
- developing extensions to deal with e.g. temporal aspects, uncertainty, vagueness, context-dependency, etc.

# Summary

- Description Logics are formalisms designed and used specifically for representing and reasoning with *terminological* and *assertional knowledge* about a domain of application.
- The crucial formal characteristic of DLs is a good balance between *expressive power* and *reasoning* capabilities.

Resources:

F. Baader, W. Nutt. Chapter 2: *Basic Description Logic*. In: F. Baader et al., The Description Logic Handbook: Theory, Implementation, and Applications, 2003.

M. Krötzsch, F. Simančik, I. Horrocks. *Description Logic Primer*, 2012.

Next:

- representation of DL knowledge bases (ontologies)
- reasoning services for DLs

▷ *Please bring laptops with Protégé ontology editor installed* `http://protege.stanford.edu/`.

▷ *Download the file* `arai-art.owl` *from the Blackboard*.