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/*/)

Szymon Klarman
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.

\[
\text{Sun} \xrightarrow{\text{is-a}} \text{Star} \xrightarrow{\text{belongs_to}} \text{Galaxy} \\
\text{Earth} \xrightarrow{\text{is-a}} \text{Planet} \xrightarrow{\text{is-a}} \text{Celestial\_Body}
\]

- 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:

\[
\begin{array}{c}
\text{First-Order Logic} \\
\text{Description Logics} \leftrightarrow \text{Modal Logics} \\
\text{propositional logic} \\
\text{(inexpressive, decidable)}
\end{array}
\]

- 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.
**ALC: Syntax**

**ALC** = Attributive Language with Complement

The *vocabulary* of a Description Logic language includes:
- *concept names*, e.g. `Man`, `Parent`, `Car` (A, B, C ...),
- *role names*, e.g. `biggerThan`, `likes`, `locatedIn` (r, s ...).

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

\[
C, D \rightarrow A \quad | \quad \text{atomic concept}
\]

\[
\top \quad | \quad \text{universal concept} \quad | \quad \text{“thing”}
\]

\[
\bot \quad | \quad \text{bottom concept} \quad | \quad \text{“nothing”}
\]

\[
\neg C \quad | \quad \text{complement} \quad | \quad \text{“not”}
\]

\[
C \cap D \quad | \quad \text{intersection} \quad | \quad \text{“and”}
\]

\[
C \cup D \quad | \quad \text{union} \quad | \quad \text{“or”}
\]

\[
\exists r . C \quad | \quad \text{existential restriction} \quad | \quad \text{“some”}
\]

\[
\forall r . C \quad | \quad \text{universal restriction} \quad | \quad \text{“only”}
\]
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

Solution:
- $\text{Artwork} \sqcap \neg \text{Sculpture}
- \exists \text{painted}.\text{Painting}
- \exists \text{sculptured}.\text{Artwork} \sqcap \forall \text{created}.\text{Sculpture}$
Exercise: modeling $ALC$ 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:
  - $\text{Artwork} \sqcap \neg \text{Sculpture}$
  - $\exists \text{painted}. \text{Painting}$
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:
  - $\text{Artwork} \sqcap \neg \text{Sculpture}$
  - $\exists \text{painted}.\text{Painting}$
  - $\exists \text{sculptured}.\text{Artwork} \sqcap \forall \text{created}.\text{Sculpture}$
**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:

\[
\begin{align*}
\top^\mathcal{I} &= \Delta^\mathcal{I} \\
\bot^\mathcal{I} &= \emptyset \\
(\neg C)^\mathcal{I} &= \Delta^\mathcal{I} \setminus C^\mathcal{I} \\
(C \cap D)^\mathcal{I} &= C^\mathcal{I} \cap D^\mathcal{I} \\
(C \cup D)^\mathcal{I} &= C^\mathcal{I} \cup D^\mathcal{I} \\
(\exists r. C)^\mathcal{I} &= \{ x \in \Delta^\mathcal{I} \mid \exists y. (x, y) \in r^\mathcal{I} \land y \in C^\mathcal{I} \} \\
(\forall r. C)^\mathcal{I} &= \{ x \in \Delta^\mathcal{I} \mid \forall y. (x, y) \in r^\mathcal{I} \rightarrow y \in C^\mathcal{I} \}
\end{align*}
\]
Exercise: semantics of $\mathcal{ALC}$ concepts

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

  \[ \text{Artwork}^\mathcal{I} = \{ \text{nightwatch, sixtChappel, thinker, david} \}, \]

  \[ \text{Artist}^\mathcal{I} = \{ \text{rembrandt, rodin, michelangelo} \} \]

  \[ \text{Sculptor}^\mathcal{I} = \{ \text{rodin, michelangelo} \} \quad \text{Sculpture}^\mathcal{I} = \{ \text{thinker, david} \} \]

  \[ \text{Painter}^\mathcal{I} = \{ \text{rembrandt, michelangelo} \} \]

  \[ \text{Painting}^\mathcal{I} = \{ \text{nightwatch, sixtChappel} \} \]

  \[ \text{painted}^\mathcal{I} = \{ (\text{rembrandt, nightwatch}), (\text{michelangelo, sixtChappel}) \} \]

  \[ \text{sculptured}^\mathcal{I} = \{ (\text{rodin, thinker}), (\text{michelangelo, david}) \} \]

  \[ \text{created}^\mathcal{I} = \{ (\text{rembrandt, nightwatch}), (\text{michelangelo, sixtChappel}), (\text{michelangelo, david}), (\text{rodin, thinker}) \} \]
Exercise: semantics of $\mathcal{ALC}$ concepts

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

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

• Solution:
  1. $(\text{Artwork} \sqcap \neg \text{Sculpture})^I = \{\text{nightwatch, sixtChappel}\}$
Exercise: semantics of ALC concepts

• Compute the semantics of the following concepts:
  1. Artwork ∩ ¬Sculpture
  2. ∃painted.Painting
  3. ∃sculptured.Artwork ∩ ∀created.Sculpture
  4. ∀created.Sculpture ∩ ∃created.(Artwork ∩ ¬Sculpture)
  5. ∀created.Painting ∩ ∃created.⊤
  6. ∃created.Painting

• Solution:
  1. (Artwork ∩ ¬Sculpture)frica = {nightwatch, sixtChappel}
  2. (∃painted.Painting)frica = {rembrandt, michelangelo}
Exercise: semantics of \textit{ALC} concepts

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

- Solution:
  1. $(\text{Artwork} \sqcap \neg \text{Sculpture})^I = \{\text{nightwatch, sixtChappel}\}$
  2. $(\exists \text{painted. Painting})^I = \{\text{rembrandt, michelangelo}\}$
  3. $(\exists \text{sculptured. Artwork} \sqcap \forall \text{created. Sculpture})^I = \{\text{rodin}\}$
Exercise: semantics of $\mathcal{ALC}$ concepts

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

- Solution:
  1. $(\text{Artwork} \sqcap \neg \text{Sculpture})^\mathcal{I} = \{ \text{nightwatch, sixtChappel} \}$
  2. $(\exists \text{painted}. \text{Painting})^\mathcal{I} = \{ \text{rembrandt, michelangelo} \}$
  3. $(\exists \text{sculptured}. \text{Artwork} \sqcap \forall \text{created}. \text{Sculpture})^\mathcal{I} = \{ \text{rodin} \}$
  4. $(\forall \text{created}. \text{Sculpture} \sqcap \exists \text{created}. (\text{Artwork} \sqcap \neg \text{Sculpture}))^\mathcal{I} = \emptyset$
Exercise: semantics of ALC concepts

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

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

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

- Solution:
  1. $(\text{Artwork} \sqcap \neg \text{Sculpture})^I = \{ \text{nightwatch, sixtChappel} \}$
  2. $(\exists \text{painted. Painting})^I = \{ \text{rembrandt, michelangelo} \}$
  3. $(\exists \text{sculptured. Artwork} \sqcap \forall \text{created. Sculpture})^I = \{ \text{rodin} \}$
  4. $(\forall \text{created. Sculpture} \sqcap \exists \text{created. } (\text{Artwork} \sqcap \neg \text{Sculpture}))^I = \emptyset$
  5. $(\forall \text{created. Painting} \sqcap \exists \text{created. } \top)^I = \{ \text{rembrandt} \}$
  6. $(\exists \text{created. Painting})^I = \{ \text{rembrandt, michelangelo} \}$
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:** \( (\neg \top)^I = \Delta I \setminus \Delta I = \emptyset = \bot^I \)

- \( \neg \bot = \top \)

- \( \neg \neg C = C \)
  
  **Proof:** \( (\neg \neg C)^I = \Delta I \setminus (\Delta I \setminus C^I) = (\Delta I \setminus \Delta I) \cup C^I = C^I \)

- \( \neg (C \cap D) = \neg C \cup \neg D \)
  
  **Proof:** \( (\neg (C \cap D))^I = \Delta I \setminus (C^I \cap D^I) = (\Delta I \setminus C^I) \cup (\Delta I \setminus D^I) = (\neg C \cup \neg D)^I \)

- \( \neg (C \cup D) = \neg C \cap \neg D \)

- \( \neg \forall r.C = \exists r.\neg C \)
  
  **Proof:** \( (\neg \forall r.C)^I = \Delta I \setminus \{x \in \Delta I \mid \forall y.(x, y) \in r^I \rightarrow y \in C^I\} = \)
  
  \( = \{x \in \Delta I \mid \neg(\forall y.(x, y) \in r^I \rightarrow y \in C^I)\} = \{x \in \Delta I \mid \exists y.(x, y) \in r^I \land y \notin C^I\} = \{x \in \Delta I \mid \exists y.(x, y) \in r^I \land y \in (\neg C)^I\} = (\exists r.\neg C)^I \)

- \( \neg \exists r.C = \forall r.\neg C \)
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:

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

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

<table>
<thead>
<tr>
<th>¬A</th>
<th>C ∩ D</th>
<th>∀r.C</th>
<th>∃r.⊤</th>
<th>¬C</th>
<th>C ∪ D</th>
<th>∃r.C</th>
<th>{a}</th>
<th>r⁻</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>AL</td>
<td>AL</td>
<td>AL</td>
<td>C</td>
<td>U</td>
<td>E</td>
<td>O</td>
<td>I</td>
</tr>
</tbody>
</table>

You can add (or remove) features from $AL$ (Attributive Language) to obtain more (or less) expressive DLs. For instance:

- $ALC = AL + C = AL + U + E$
- $EL = AL - (∀r.C) - (¬A) + E$
- $SROIQ(D) = \text{all above and more}$
Expressiveness vs. complexity

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

<table>
<thead>
<tr>
<th>DL</th>
<th>complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{EL}$</td>
<td>PTIME</td>
</tr>
<tr>
<td>$\mathcal{ALC}$</td>
<td>ExpTime-complete</td>
</tr>
<tr>
<td>$\cdots$</td>
<td>$\cdots$</td>
</tr>
<tr>
<td>$\mathcal{SROIQ}(D)$</td>
<td>N2ExpTime-complete</td>
</tr>
</tbody>
</table>


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 ;)

Szymon Klarman
Relationships to other logics

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

<table>
<thead>
<tr>
<th>DL</th>
<th>FOL</th>
<th>Modal Logic</th>
<th>Propositional Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$A(x)$</td>
<td>$p_A$</td>
<td>$p_A$</td>
</tr>
<tr>
<td>$r$</td>
<td>$r(x, y)$</td>
<td>access. relation $r$</td>
<td>inexpressible</td>
</tr>
<tr>
<td>$\exists r. A$</td>
<td>$\exists y. (r(x, y) \land A(y))$</td>
<td>$\Diamond r p_A$</td>
<td>inexpressible</td>
</tr>
</tbody>
</table>

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:

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.