An Introduction to Description Logic I

Introduction and Historical remarks

Marco Cerami

Palacký University in Olomouc
Department of Computer Science
Olomouc, Czech Republic

Olomouc, October 30th 2014
Introduction
What are Description Logics?

- Description Logics (DLs) are knowledge representation languages.

- The general framework they belong to, is Knowledge Representation and Reasoning (KR) in Artificial Intelligence (AI).

- DLs are strongly based on Formal Logic, in particular they can be seen as a fragment of First Order Logic or a notational variant of Modal Logic.

- They main effort of the research on DLs is characterized by the search of a fair trade-off between expressivity and computational complexity.
Representing Knowledge and Reasoning with it
Conceptual Representation

DLs are used to represent concepts:

Person ⊓ Female
“female person”

Person ⊓ ∀ hasChild. Male
“person who has only sons (if he has children)”

and their relations

Person ⊓ Male ⊑ Person
“every male person is a person”
Inductive definition of new concepts

There is no need of storing a huge number of primitive definitions:

Person, Female, Male, FemalePerson, MalePerson, NonPerson, NonFemale, NonMale, FemaleOrMale, NonFemalePerson, NonMalePerson, etc..

DLs concepts are formed \textit{inductively} from relatively few building blocks:

From

Person, Female, Male,

Build

Female \sqcap \text{Person}, \text{Male} \sqcap \text{Person}, \neg \text{Person}, \neg \text{Female}, \neg \text{Male},

Female \sqcap \text{Male}, \neg (\text{Female} \sqcap \text{Person}), \neg (\text{Male} \sqcap \text{Person}), \text{etc.}
Binary relations between individuals

The language of DL, admits the use of **roles**, that represent binary relations between individuals.

\[
\text{Person} \sqcap \text{Male} \sqcap \exists \text{hasChild}(\text{Female} \sqcap \text{Person})
\]

“man who has a daughter”

\[
\text{Person} \sqcap \text{Female} \sqcap \exists \text{hasChild}(\text{Male} \sqcap \text{Person})
\]

“woman who has a son”

thus creating different layers for concepts.
Reasoning: (in)consistency

The main strength of DLs is **reasoning** with concepts, e.g. to prove **consistency**, like:

\[
\text{Person} \sqsubseteq \forall \text{hasChild}.\text{Male}
\]

or inconsistency:

\[
\text{Person} \sqsubseteq \forall \text{hasChild}.\text{Male} \sqcap \exists \text{hasChild}.(\text{Person} \sqcap \neg \text{Male})
\]
Reasoning: inference

DLs can be used to infer **hidden information** from existing knowledge:

From

\[ \text{Female} \sqcap \text{Male} \sqsubseteq \bot \quad \text{and} \quad \forall \text{hasChild}. \text{Male}(\text{John}) \]

Infer

\[ \neg \exists \text{hasChild}. \text{Female}(\text{John}) \]

or

\[ \text{Female} \sqsubseteq \neg \text{Male} \]
Logical reasoning vs rules

There is no need of storing a huge number of rules:

\[
\text{Female} \sqcap \text{Male} \sqsubseteq \bot, \\
\forall \text{hasChild}. \text{Female} \sqcap \forall \text{hasChild}. \text{Male} \sqsubseteq \bot, \\
\forall \text{hasChild}. \text{Female} \sqcap \exists \text{hasChild}. \text{Male} \sqsubseteq \bot, \\
\exists \text{hasChild}. \text{Female} \sqcap \forall \text{hasChild}. \text{Male} \sqsubseteq \bot, \\
\forall \text{hasChild}. \forall \text{hasChild}. \text{Feale} \sqcap \forall \text{hasChild}. \forall \text{hasChild}. \text{Male} \sqsubseteq \bot, \\
\forall \text{hasChil}. \forall \text{hasChild}. \text{Feale} \sqcap \forall \text{hasChil}. \forall \text{hasChild}. \text{Male} \sqsubseteq \bot, \\
\forall \text{hasChil}. \forall \text{hasChild}. \text{Feale} \sqcap \forall \text{hasChil}. \exists \text{hasChild}. \text{Male} \sqsubseteq \bot, \\
\exists \text{hasChild}. \forall \text{hasChild}. \text{Feale} \sqcap \forall \text{hasChild}. \exists \text{hasChild}. \text{Male} \sqsubseteq \bot, \\
e \text{tc...}
\]

since they can be all inferred from

\[
\text{Female} \sqcap \text{Male} \sqsubseteq \bot,
\]
Historical Remarks
The origins of DL systems

- Description Logics are the **result** of at least 30 years of research on the field of knowledge representation.

- This research did not begin within the DL framework, rather it started from researches about human **cognitive behavior**.

- It arrived to this logic-based framework through an **evolution process** of older formalisms such as:
  
  - Frame-based systems,
  
  - KL-ONE based systems.
Frame-based systems

- Frame-based systems were formalisms based on researches about human cognitive behavior.
- They were systems based on the old idea that human mind can be represented in its totality by a more or less comprehensive program.
- In this sense, their goal was to obtain a program that imitates human mental skills, e.g. natural language understanding.
- For this reason these systems were thought in such a way that they could support language ambiguity.
- For those fact these old systems were far from being based on formal logic, when their authors were not explicitly against the use of logic.
- The main examples of frame-based systems are
  - Quillian’s Semantic networks
  - Minsky’s Frame systems.
Semantic networks

- **Semantic networks** (60’s-70’s) have been defined with the aim of giving an account of the way human memory works.

- A program is defined, that can be roughly divided into three parts:
  - The first part is a **memory model** that works like a linked vocabulary.
  - The second part of the program is a **search program** and allows to look for hidden relations between words.
  - The third part of the program is a **sentence generator**, which utilizes the work done by the search program to express sentences in natural language.
The memory model

Frame Systems

- **Frame systems** (70’s-80’s) have been defined with the aim of explaining the way people face known challenges by using **mental frames**.
- Frames are data structures that represent **stereotyped situations**.
- At the higher levels of a frame there are nodes that do not change with the **instantiation of a situation**.
- At the lower levels there are **empty nodes that can be filled up** either with contingent information or with other frames.
- People use mental frames to **act fast**.
- When either a **new situation** is faced, preexisting frames are either modified or substituted by new ones.
- Minsky’s frame systems are often considered an example of **default reasoning**.
Features of Frame Systems

- Formally a frame system is a set of frames that consider the same situation seen from different points of view.

- Among the reasoning services of frame systems there are:
  1. subsumption between frames, in order to give specific situations a more general meaning,
  2. search of slot fillers, in order to add information to a given situation.

- there is no standard semantics,

- a number of expert systems based on this formalism have been done.
Example of KEE Knowledge Base

Frame: Course in KB University
  MemberSlot: enrolls
  ValueClass: Student
  Cardinality.Min: 2
  Cardinality.Max: 30
  MemberSlot: taughtby
  ValueClass: (UNION GradStudent Professor)
  Cardinality.Min: 1
  Cardinality.Max: 1

Frame: AdvCourse in KB University
  SuperClasses: Course
  MemberSlot: enrolls
  ValueClass: (INTERSECTION GradStudent (NOT Undergrad))
  Cardinality.Max: 20

Frame: BasCourse in KB University
  SuperClasses: Course
  MemberSlot: taughtby
  ValueClass: Professor

Frame: Professor in KB University

Frame: Student in KB University

Frame: GradStudent in KB University
  SuperClasses: Student
  MemberSlot: degree
  ValueClass: String
  Cardinality.Min: 1
  Cardinality.Max: 1

Frame: Undergrad in KB University
  SuperClasses: Student

Limits of Frame-based systems

During the second half of 70’s began to be clear the limitations of frame-based systems. Among those limitations we can find the following ones:

- it was not so clear what the systems had to compute,
- the semantics of procedural aspects was not very clear,
- there was no simple way to give these systems a clear formal semantics,
- despite these formalism were presented as an alternative to logic-based formalisms, most aspects of these systems could be formalized by means of first order logic.
KL-ONE

KL-ONE is a knowledge representation system developed since 1979 with the following features:

- it considers the tasks of extracting implicit conclusions from existing knowledge,
- it gives the user the possibility of defining new complex concepts and roles,
- it introduces the difference between individual concepts and generic concepts,
- the difference between the concept definitions with sufficient and necessary condition and those with just necessary ones is studied,
- are added to the reasoning tasks:
  - classification (computation of the hierarchy of subsumptions),
  - realization (computation of the more specific atomic concept).
Limits of KL-ONE

Besides these novelties, KL-ONE had some weaknesses that became evident quite early.

- The lack of a clear formal semantics.
- **Algorithms** for deciding classification and realization were incomplete.
- The system was thought under the point of view of the mere concept representation, more than functionality.
- The lack of a clear distinction between the knowledge representing relations among concepts and that representing assertions about individuals.

Some of these shortcoming are taken into account to build further KL-ONE-based systems.
A new framework

The KL-ONE experience brought a new way to see knowledge representation systems.

- it has been adopted the so-called functional approach.
- This is at the origin of the growing interest on decision algorithms and their complexity.
- The need of a clear semantics can be seen at the origin of the fact that systems began to be more and more logic-based.
- This allowed to think about those systems in a more abstract way as clearly defined description languages.
- The languages are now quantitatively comparable, mainly under two points of view:
  - the computational complexity of reasoning,
  - the expressivity of the language.
Applications
The Semantic Web

- **The Semantic Web** is the effort to provide a common framework that allows data to be shared and reused across applications, enterprises and communities.

- The aim is to create or select a common format for integration and combination of data coming from different sources.

- The idea is that the common format for different source data can be given by the real world objects data relates to.

- Data are related to real world objects through **ontologies**.

- A language that is commonly used to express ontologies is **OWL 2**.
Ontologies

- An **Ontology** is a set of precise descriptive statements about a domain of interest.

- The aim is to make software **behave in a uniform way** and work well with other software.

- The idea is that the meaning of a term can be characterized by its **interrelations to other terms**.

- E.g., the term “son” can be characterized by its interrelations to the terms “parent”, “grandson”, “brother”, “sister”, etc.

- The term “son’ can be translated into “syn” because this word has the same interrelations to the terms “rodiče”, “vnuček”, “bratr”, “sestra”, etc.
OWL 2

- **OWL 2** (Web Ontology Language) is a language for expressing ontologies.
- Names in OWL 2 are **international resource identifiers** (IRIs).
- Some examples\(^1\) of OWL 2 axioms:

```xml
<SubClassOf>
  <Class IRI="Mother"/>
  <Class IRI="Woman"/>
</SubClassOf>

<DisjointClasses>
  <Class IRI="Woman"/>
  <Class IRI="Man"/>
</DisjointClasses>
```

\(^1\)The examples are taken from *OWL 2 Web Ontology Language Primer (Second Edition)*, Edited by P. Hitzler et al.
Reasoners and DLs

- Clearly, the “knowledge” that can be represented by OWL 2 does not reflect all aspects of human knowledge.

- But the knowledge that can be represented in OWL 2 can be managed in an efficient way.

- An OWL 2 ontology can be indeed managed by some reasoner, that computes the consequences of the knowledge stored in a ontology using OWL 2 and infers hidden knowledge.

- The role of DLs lies in the formal study of the logical and computational features of “subsets” of OWL 2. In particular:
  - the design of suitable algorithms for reasoners,
  - the trade-off between expressivity and complexity.
Medical ontologies

Other examples of application of DLs are medical ontologies:

- **OpenGALEN** (not active anymore):
  http://www.opengalen.org

- **SNOMED CT**:

Here, again, DLs are used as the underlying formalism of the search engines.