An Introduction to Description Logics

Marco Cerami

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

Olomouc, 21.11.2013

INVESTMENTS IN EDUCATION DEVELOPMENT

Introduction

What are Description Logics?

- Description Logics (DLs) are logic-based **knowledge representation languages**.
- The general framework they belong to, is **Knowledge Representation and Reasoning** (KR) in **Artificial Intelligence** (AI).
- They are characterized by the search of a **fair trade-off** between **expressivity** and computational **complexity** in KR.
- Some **examples of their application** are:
  - as the underlying formalism for the **Semantic Web**;
  - as search engine for **knowledge bases** (e.g. GALENO).

Aims of Description Logics

The aims of DLs is twofold:

- they are used to **represent concepts** and their relations **beyond the super-sub-concept relation**:

$$\texttt{Person} \sqcap \texttt{Female}$$

"female person"

$$\texttt{Person} \sqcap \forall \texttt{hasChild}.\texttt{Male}$$

"person who has only sons (if he has children)"

- they are used to **reason** with them, e.g.
  - to prove **(in)consistency** of concepts, like:

$$\texttt{Person} \sqcap \forall \texttt{hasChild}.\texttt{Male} \sqcap \forall \texttt{hasChild}.(\texttt{Person} \sqcap \texttt{Female})$$

  - to infer **hidden information** from existing knowledge.

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.
- This research did not begin within the DL framework, rather arrived to this framework through an **evolution process** of older formalisms such as:
- 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

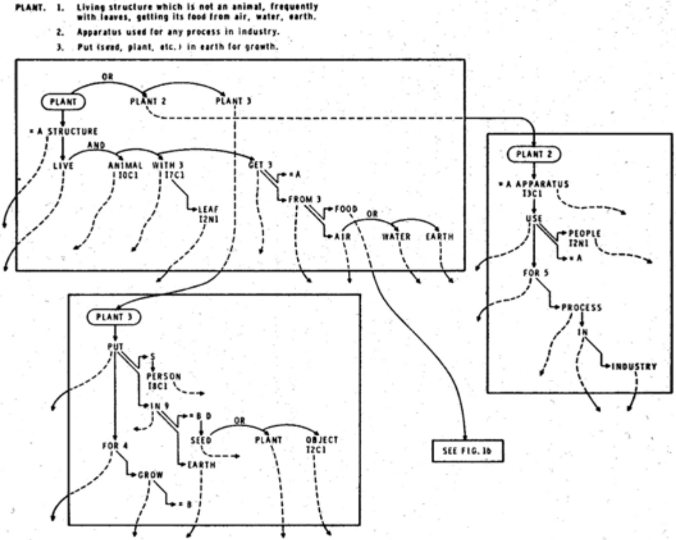


Fig. 1a. Three Planes Representing Three Meanings of "Plant."

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**.
- The fact that the **algorithms** for deciding classification and realization were **incomplete**.
- The fact of thinking the system 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.

Syntax

Description Signature

A **description signature** is a tuple $\mathbf{D} = \langle N_I, N_A, N_R \rangle$, where

- N_I, a set of **individual names**;
  - Notation: $a, b, c, \ldots$
  - Examples: `John`, `Mary`, `Prague`, `MainSquare`,
- N_A a set of concept names (the **atomic concepts**);
  - Notation: $A, B, C, \ldots$
  - Examples: `Person`, `Female`, `Tall`, `Fat`, `Hight`,
- N_R a set of role names (the **atomic roles**)
  - Notation: $R_1, R_2, \ldots$
  - Examples: `hasChild`, `hasSister`, `hasNear`, `hasTemperature`,

Complex Concepts

$C, D \longrightarrow$

$\bot$	empty concept	$\mathcal{FL}_0$
$\top$	universal concept	$\mathcal{FL}_0$
A	atomic concept	$\mathcal{FL}_0$
$C \sqcap D$	conjunction	$\mathcal{FL}_0$
$\forall R.C$	value restriction	$\mathcal{FL}_0$
$\exists R.\top$	restricted existential quantif.	$\mathcal{FL}^-$
$\neg A$	atomic complementation	$\mathcal{AL}$
$\neg C$	complementation	$\mathcal{C}$
$C \sqcup D$	disjunction	$\mathcal{U}$
$\exists R.C$	existential quantification	$\mathcal{E}$
$\geq n\, R$	unqualified	
$\leq n\, R$	number	$\mathcal{N}$
$= n\, R$	restriction	
$\geq n\, R.C$	qualified	
$\leq n\, R.C$	number	$\mathcal{Q}$
$= n\, R.C$	restriction	
$\{a\}$	nominals	$\mathcal{O}$
d	concrete domains	(D)

Languages

- The name $\mathcal{FL}$ stands for **frame language** because it has more or less the same expressive power of frame-based systems; it was studied in the 80's;
- the name $\mathcal{AL}$ stands for **attributive language**, began to be studied in the last 80's;
- $\mathcal{AL}$ marks the difference between frame-based systems and the new systems based on a **description of attributes and predicates**;
- a central role has been played in the 90's by the language $\mathcal{ALC}$ because it is **the most related** to modal and predicate **logic**.

Role-based languages

There are other languages that are defined by the behavior of role constructors from $\mathcal{ALC}$:

R, S	$\longrightarrow$	R	atomic role	$\mathcal{FL}_0$
		R^+	transitive role	$\mathcal{S}$
		U	universal role	$\mathcal{S}$
		R^-	inverse role	$\mathcal{I}$
		$R \sqcap S$	role intersection	$\mathcal{H}$
		$\neg R$	role complementation	$\mathcal{H}$
		$R \sqcup S$	role union	$\mathcal{H}$
		$R \circ S$	role composition	$\mathcal{R}$
		f	functional role (feature)	$\mathcal{F}$

Semantics

Interpratations

An **interpretation** is a pair

$$\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$$

where:

- $\Delta^{\mathcal{I}}$ is a nonempty set, called **domain**;
- $\cdot^{\mathcal{I}}$ is an **interpretation function** that assigns:
  - to each individual name $a \in N_I$ an element

    $$a^{\mathcal{I}} \in \Delta^{\mathcal{I}},$$

  - to each atomic concept A a subset of the domain set

    $$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}},$$

  - to each role name R a binary relation on the domain set

    $$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}.$$

Semantics of complex concepts

$$
\begin{aligned}
\bot^{\mathcal{I}} &= \emptyset \\
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\exists R.\top)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} : \text{exists } w \in \Delta^{\mathcal{I}} \text{ such that } R^{\mathcal{I}}(v, w)\} \\
(\forall R.C)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} : \text{for every } w \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(v, w) \rightarrow C^{\mathcal{I}}(w)\} \\
(\exists R.C)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} : \text{exists } w \in \Delta^{\mathcal{I}} \text{ s. t. } R^{\mathcal{I}}(v, w) \wedge C^{\mathcal{I}}(w)\} \\
(\geq n\, R)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} : |\{b \in \Delta^{\mathcal{I}} : R^{\mathcal{I}}(a, b)\}| \geq n\} \\
(\leq n\, R)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} : |\{b \in \Delta^{\mathcal{I}} : R^{\mathcal{I}}(a, b)\}| \leq n\} \\
(= n\, R)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} : |\{b \in \Delta^{\mathcal{I}} : R^{\mathcal{I}}(a, b)\}| = n\} \\
(\geq n\, R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} : |\{b \in \Delta^{\mathcal{I}} : R^{\mathcal{I}}(a, b) \wedge C^{\mathcal{I}}(b)\}| \geq n\} \\
(\leq n\, R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} : |\{b \in \Delta^{\mathcal{I}} : R^{\mathcal{I}}(a, b) \wedge C^{\mathcal{I}}(b)\}| \leq n\} \\
(= n\, R.C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} : |\{b \in \Delta^{\mathcal{I}} : R^{\mathcal{I}}(a, b) \wedge C^{\mathcal{I}}(b)\}| = n\} \\
\{a\}^{\mathcal{I}} &= \{a^{\mathcal{I}}\} \subseteq \Delta^{\mathcal{I}}
\end{aligned}
$$

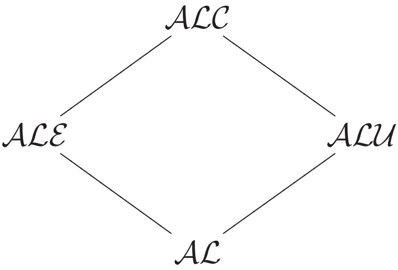
Semantics of complex roles

$$
\begin{aligned}
U^{\mathcal{I}} &= \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \\
(R^{-})^{\mathcal{I}} &= \{(b, a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \colon (a, b) \in R^{\mathcal{I}}\} \\
(\neg R)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}} \\
(R \sqcap S)^{\mathcal{I}} &= R^{\mathcal{I}} \cap S^{\mathcal{I}} \\
(R \sqcup S)^{\mathcal{I}} &= R^{\mathcal{I}} \cup S^{\mathcal{I}} \\
(R \circ S)^{\mathcal{I}} &= R^{\mathcal{I}} \circ S^{\mathcal{I}}
\end{aligned}
$$

The semantics of transitive, reflexive and functional roles is the usual for transitive and reflexive relations or functions.

Inclusions between languages: the $\mathcal{ALC}$ hierarchy

A straightforward consequence of the semantics of constructors is that every $\mathcal{ALE}$ and every $\mathcal{ALU}$ concepts are $\mathcal{ALC}$ concepts, but there are $\mathcal{ALE}$ concepts that are not $\mathcal{ALU}$ concepts and vice-versa. So, the hierarchy of languages between $\mathcal{AL}$ and $\mathcal{ALC}$ appears as follows



Knowledge Bases

Axioms

- An **inclusion axiom** is an expression of the form:

$$C \sqsubseteq D$$

  where C, D are concepts.

- An **assertion axiom** is an expression of the form:

$$C(a)$$

  where C is concept and a is an individual.

- A **role axiom** is an expression of the form:

$$R \sqsubseteq S$$

  where R, S are roles.

Semantics of axioms

- The inclusion axiom $C \sqsubseteq D$ is **true** iff **for every interpretation** $\mathcal{I}$:

$$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}.$$

- The assertion axiom $C(a)$ is **true** iff **for every interpretation** $\mathcal{I}$:

$$a^{\mathcal{I}} \in C^{\mathcal{I}}.$$

- The role axiom $R \sqsubseteq S$ is **true** iff **for every interpretation** $\mathcal{I}$:

$$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}.$$

Knowledge Bases

- A **terminological box** (TBox) is a finite set of inclusion axioms.
- An **assertional box** (ABox) is a finite set of assertion axioms.
- A **relational box** (RBox) is a finite set of role axioms.
- An **Knowledge Base** (KB) is a triple

$$\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{R})$$

  where $\mathcal{T}$ is a TBox, $\mathcal{A}$ is an ABox and $\mathcal{R}$ is an RBox (each one possibly empty).

Reasoning Tasks

Reasoning tasks

Consider a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathcal{R})$, a pair of concepts C, D and an individual a, then we can define the main reasoning tasks considered in the literature.

- $\mathcal{K}$ is **consistent** when there is an interpretation $\mathcal{I}$ that satisfies every axiom in $\mathcal{K}$. In symbols $\mathcal{I} \models \mathcal{K}$.
- C is **satisfiable** with respect to the (possibly empty) knowledge base $\mathcal{K}$ when there exists an interpretation $\mathcal{I}$ satisfying $\mathcal{K}$, such that $C^{\mathcal{I}} \neq \emptyset$.
- D **subsumes** concept C with respect to the (possibly empty) knowledge base $\mathcal{K}$ when, in every interpretation $\mathcal{I}$ that satisfies $\mathcal{K}$, it holds that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Iin symbols $\mathcal{K} \models C \sqsubseteq D$.
- An axiom φ (either inclusion or assertion) is **entailed** by a knowledge base $\mathcal{K}$ (in symbols $\mathcal{K} \models \varphi$) when, in every model $\mathcal{I}$ of $\mathcal{K}$, it holds that $\varphi^{\mathcal{I}} = 1$.

Reduction to knowledge base consistency

Each one of the above reasoning problems can be **reduced to knowledge base (in)consistency** in the following way:

- Concept C is **satisfiable with respect to the knowledge base** $\mathcal{K}$ if and only if the new knowledge base $\mathcal{K} \cup \{C(a)\}$ is consistent, where a is an individual name not occurring in $\mathcal{K}$.
- Concept D **subsumes concept** C **with respect to the knowledge base** $\mathcal{K}$ if and only if the new knowledge base $\mathcal{K} \cup \{(C \sqcap \neg D)(a)\}$ is inconsistent, where a is a new individual name.
- An axiom φ (either inclusion or assertion) is **entailed** by a knowledge base $\mathcal{K}$ if and only if the new knowledge base $\mathcal{K} \cup \{\neg\varphi\}$ is inconsistent. Here $\neg\varphi = \neg C(a)$, if $\varphi = C(a)$ and $\neg\varphi = C \sqcap \neg D(a)$, for a new individual name a, if $\varphi = C \sqsubseteq D$.

Complexity

Complexity: classical results

The study of the computational complexity of the reasoning tasks is fundamental in Description Logics. Some classical results are:

- **subsumption with respect to empty KBs** in language $\mathcal{FL}^-$ is in P [Brachman and Levesque, 1983],
- **concept satisfiability with respect to empty KBs**, in language $\mathcal{ALU}$ is co-NP [Schmidt-Schauss and Smolka 1991],
- **concept satisfiability with respect to empty KBs**, in language $\mathcal{ALE}$ is NP [Donini et al. 1992],
- **concept satisfiability with respect to empty KBs** in language $\mathcal{ALC}$ is PSPACE-complete [Schmidt-Schauss and Smolka 1991],
- **Knowledge base consistency** for language $\mathcal{ALC}$ is in EXPTIME [Donini and Masacci 2000].

Complexity: further results

	Sat.	Unsat.	Sat. acyclic KB	Sat. w.r.t. KB	Subs.
$\mathcal{FL}^-$					PTIME
$\mathcal{AL}$			co-NP	EXPTIME	PTIME
$\mathcal{ALI}$					PTIME
$\mathcal{ALN}$			PSPACE		PTIME
$\mathcal{ALNI}$	PTIME				co-NP
$\mathcal{ALE}$		NP	co-NP	PSPACE	NP
$\mathcal{FL}^-\mathcal{E}$		NP			NP
$\mathcal{ALR}$		NP			NP
$\mathcal{ALER}$		NP			NP
$\mathcal{ALU}$		co-NP			co-NP
$\mathcal{ALC}$	PSPACE	PSPACE			PSPACE
$\mathcal{ALEN}$	PSPACE				
$\mathcal{ALUR}$	PSPACE				
$\mathcal{ALNR}$	PSPACE				
$\mathcal{ALCNR}$	PSPACE				
$\mathcal{ALCH}$	NEXPTIME				
$\mathcal{ALCNO}$	NEXPTIME				
$\mathcal{ALCNR}$				NEXPTIME	

Sources of indeterminism

For many languages, often a systematic study of **what causes the increase of complexity** has been undertaken. Some examples of those systematic studies are:

- **subsumption** in language $\mathcal{FL}^-$ jumps from P to co-NP when a TBox is considered [Nebel, 1990],
- **concept satisfiability with respect to empty KBs**, in language $\mathcal{FL}^-$ jumps from P to co-NP when disjunction and atomic complementation are added [Schmidt-Schauss and Smolka 1991],
- **concept satisfiability with respect to empty KBs** in language $\mathcal{FL}^-$ jumps from P to PSPACE when unrestricted complementation is added [Schmidt-Schauss and Smolka 1991],
- **concept satisfiability with respect to empty KBs** in language $\mathcal{FL}^-$ jumps from P to NP when unrestricted existential quantification is added [Donini et al. 1992].

Algorithms

- The DL systems of the 80's used so-called **structural subsumption algorithms**:
  - perform a **comparison in the syntactic structure** of two given concepts in a suitable normal form;
  - relatively efficient when applied to very **inexpressive languages**;
  - in more expressive languages are **incomplete**.
- The 90's saw the introduction of the **tableau based algorithms**:
  - **complete** also for quite expressive DLs;
  - allowed a **systematic study of complexity of reasoning** in different DLs;
  - suitable to be **highly optimized**.

Thank you for the attention !