An Introduction to Description Logic II

Syntax and Semantics

Marco Cerami

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

Olomouc, October 30th 2014

european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
INVESTMENTS IN EDUCATION DEVELOPMENT

Syntax

General Remarks

- DL languages differ for the presence of **concept constructors** $\sqcap$, $\sqcup$, $\neg$, etc..
- Each DL language is identified by a **name** $\mathcal{AL}$, $\mathcal{ALE}$, $\mathcal{ALC}$, etc. that denotes the set of concept constructors present in the language.
- A language $\mathcal{L}'$ is said to be **more expressive** than a language $\mathcal{L}''$, if in $\mathcal{L}'$ can be expressed every concept expressible in $\mathcal{L}''$.
- If a language $\mathcal{L}'$ **syntactically expands** a language $\mathcal{L}''$, clearly it is more expressive than $\mathcal{L}''$.
- Nevertheless, due to **semantics** reasons, there are languages that are more expressive than others, even though they are not syntactical expansions.

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** (or atomic concepts);
  - Notation: $A, B, C, \ldots$
  - Examples: `Person`, `Female`, `Tall`, `Fat`, `Hight`,
- N_R a set of **role names** (or atomic roles)
  - Notation: $R_1, R_2, \ldots$
  - Examples: `hasChild`, `hasSister`, `hasNear`, `hasTemperature`,

Frame Languages

The name $\mathcal{FL}$ stands for **frame language** because it has more or less the same expressive power of frame-based systems; they were studied in the 80's.

C, D	$\longrightarrow$	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.	$\mathcal{FL}^-$

Examples

An example of $\mathcal{FL}_0$ concept:

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

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

An example of $\mathcal{FL}^-$ concept:

$$\texttt{Person} \sqcap \exists \texttt{hasChild}.\top$$

"person who has a child"

Attributive Languages

- 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**;
- Attributive Languages are the **expansions of $\mathcal{FL}^-$** by means of the following constructors:

C, D	$\longrightarrow$	$\neg A$	atomic complementation	$\mathcal{AL}$
		$C \sqcup D$	disjunction	$\mathcal{ALU}$
		$\exists R.C$	existential quantification	$\mathcal{ALE}$
		$\neg C$	complementation	$\mathcal{ALC}$

Examples

An example of $\mathcal{AL}$ concept:

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

"person who has no sons"

An example of $\mathcal{ALU}$ concept:

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

"person who has either only sons or only daughters"

Examples

An example of $\mathcal{ALE}$ concept:

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

"person who has a son"

An example of $\mathcal{ALC}$ concept:

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

"person who has not only sons"

Existential Languages

- The name $\mathcal{EL}$ stands for **Existential Language**;
- The interest for $\mathcal{EL}$ began in the **last ten years**;
- this interest is due to its **good computational behavior** and the fact that some **ontology can be defined** in $\mathcal{EL}$.

C, D	$\longrightarrow$	A	atomic concept	$\mathcal{EL}$
		$C \sqcap D$	disjunction	$\mathcal{EL}$
		$\exists R.C$	existential quantification	$\mathcal{EL}$
		$C \sqcup D$	disjunction	$\mathcal{ELU}$
		$\neg C$	complementation	$\mathcal{ELC}$

Examples

An example of $\mathcal{EL}$ concept:

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

"person who has a son"

An example of $\mathcal{ELU}$ concept:

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

"person who has either a son or a daughter"

An example of $\mathcal{ELC}$ concept:

$$\texttt{Person} \sqcap \neg \exists \texttt{hasChild.Male}$$

"person who has no son"

Cardinality restrictions

Also some kinds of **cardinality restrictions** on the range of roles are considered:

C, D	$\longrightarrow$	$\geq n\, R$	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	

Examples

An example of $\mathcal{ALN}$ concept:

$$\texttt{Person} \sqcap (\geq 3)\texttt{hasChild} \sqcap \forall\texttt{hasChild.Male}$$

"person who has at least three children and has only sons"

An example of $\mathcal{ALQ}$ concept:

$$\texttt{Person} \sqcap (\geq 3)\texttt{hasChild.Male}$$

"person who has at least three sons"

Other languages

Other languages **built up from concept constructors** are the ones obtained by adding constructors for:

- **nominals**,
- **concrete domains**.

C, D	$\longrightarrow$	$\{a_1, \ldots, a_m\}$	nominals	$\mathcal{O}$
		d	concrete domains	(D)

Examples

An example of $\mathcal{ALO}$ concept:

$$\texttt{Country} \sqcap \forall \texttt{hasBorderWith}\{\texttt{Germany},\texttt{Poland},\texttt{Slovakia},\texttt{Austria}\}$$

"Country which has borders only with Germany, Poland, Slovakia and Austria"

An example of $\mathcal{EL}(\mathbb{N})$ concept:

$$\texttt{Person} \sqcap \exists \texttt{hasAge.30} \sqcap \exists \texttt{hasChild.}(\texttt{Male} \sqcap \exists \texttt{hasAge.10})$$

"person who is 30 years old and has a 10 years old son"

Role-based languages

There are other languages that are defined by **modifying the behavior** of role:

- The language $\mathcal{S}$ is obtained from $\mathcal{ALC}$ by allowing **transitive roles**.
- If $\mathcal{L}$ is a language, then the language $\mathcal{LH}$ is obtained by allowing **inclusion axioms** between roles.
- If $\mathcal{L}$ is a language, then the language $\mathcal{LI}$ is obtained by allowing **inverse roles** R^{-}.
- If $\mathcal{L}$ is a language, then the language $\mathcal{LF}$ is obtained by allowing **functional roles**.
- If $\mathcal{L}$ is a language, then the language $\mathcal{LR}$ is obtained by allowing **intersection of roles** $R \circ P$, sometimes along with **conjunction**, **disjunction** and **negation** of roles.

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

Set theoretical fragment

The fragment of DL languages with just **propositional connectives** can be managed with set-theoretical basic operations on the domain:

$$
\begin{aligned}
\bot^{\mathcal{I}} &= \emptyset \\
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}}
\end{aligned}
$$

Modal fragment

The fragment of DL languages with **quantifiers** can be managed with modal-like semantics:

$$
\begin{aligned}
(\exists R.\top)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} \colon \text{exists } w \in \Delta^{\mathcal{I}} \text{ such that } R^{\mathcal{I}}(v, w)\} \\
(\forall R.C)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} \colon \text{for every } w \in \Delta^{\mathcal{I}}, \\
&\quad \text{if } R^{\mathcal{I}}(v, w) \text{ then } C^{\mathcal{I}}(w)\} \\
(\exists R.C)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} \colon \text{exists } w \in \Delta^{\mathcal{I}} \text{ such that} \\
&\quad R^{\mathcal{I}}(v, w) \text{ and } C^{\mathcal{I}}(w)\}
\end{aligned}
$$

Bounded quantifiers

The concept constructors denoting **cardinality restrictions** can be managed with bounded quantifiers:

$$
\begin{aligned}
(\geq n\, R)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} \colon |\{b \in \Delta^{\mathcal{I}} \colon R^{\mathcal{I}}(v, w)\}| \geq n\} \\
(\leq n\, R)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} \colon |\{b \in \Delta^{\mathcal{I}} \colon R^{\mathcal{I}}(v, w)\}| \leq n\} \\
(= n\, R)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} \colon |\{b \in \Delta^{\mathcal{I}} \colon R^{\mathcal{I}}(v, w)\}| = n\} \\
(\geq n\, R.C)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} \colon |\{b \in \Delta^{\mathcal{I}} \colon R^{\mathcal{I}}(v, w) \wedge C^{\mathcal{I}}(w)\}| \geq n\} \\
(\leq n\, R.C)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} \colon |\{b \in \Delta^{\mathcal{I}} \colon R^{\mathcal{I}}(v, w) \wedge C^{\mathcal{I}}(w)\}| \leq n\} \\
(= n\, R.C)^{\mathcal{I}} &= \{v \in \Delta^{\mathcal{I}} \colon |\{b \in \Delta^{\mathcal{I}} \colon R^{\mathcal{I}}(v, w) \wedge C^{\mathcal{I}}(w)\}| = n\}
\end{aligned}
$$

Other concept constructors

- The semantics of **nominal concepts** is the following:

$$\{a_1, \ldots, a_m\}^{\mathcal{I}} = \{a_1^{\mathcal{I}}, \ldots, a_m^{\mathcal{I}}\} \subseteq \Delta^{\mathcal{I}}$$

- **Concrete domains** allow to consider data coming from a domain D different from the abstract domain $\Delta^{\mathcal{I}}$. Usually the sets $\mathbb{N}$ or $\mathbb{R}$ are considered, with their **natural semantics**. Indeed, the use of concrete domains is due to the fact that they usually have their own semantics and there is **no need of an ontology** to state the relations among the domain elements.

Semantics of complex roles

The definition of a concept-like semantics for role constructor is possible just for the following constructors:

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

Restrictions on roles

The semantics of **transitive** and **functional** roles is given by means of general restrictions:

- a role R is **transitive** if $R \circ R$ is equal to R,
- A role R is **functional** if

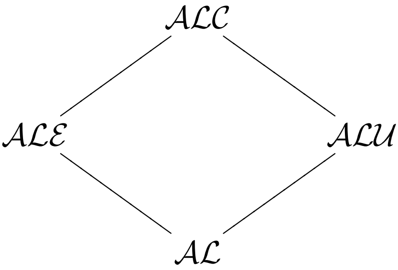
$$\forall v, w, z \in \Delta^{\mathcal{I}}, \quad R^{\mathcal{I}}(v, w) \wedge R^{\mathcal{I}}(v, z) \quad \Longrightarrow \quad w = z$$

Role hierarchies just admit inclusion axioms between roles.

Language Hierarchies

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



Other inclusions

Let $\mathcal{L}$ a DL language, then:

- language $\mathcal{LQ}$ includes language $\mathcal{LN}$:

$$(\geq n)R.\top$$

- language $\mathcal{LQ}$ includes language $\mathcal{LF}$:

$$(\leq 1)R.C$$

- **role composition** can be defined in $\mathcal{EL}$

$$\exists(R \circ P).C \equiv \exists R.\exists P.C$$