An Introduction to Description Logic IV

Relations to first order logic

Marco Cerami

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

Olomouc, November 6^{th} 2014

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

Preliminaries:

First order logic

Syntax: signature and terms

A **predicate signature s** consists of:

- a countable set of **predicate symbols** $P_1, \dots, P_n, \dots$, each one with arity ≥ 1,
- a countable set of **function symbols** $f_1, \dots, f_n, \dots$, each one with its arity,
- a countable set of **constant symbols** $c_1, \dots, c_n, \dots$, that are 0-ary function symbols.

Given a countable set *Var* of individual variables, the set of **Terms** over a predicate signature is defined inductively as follows:

- every variable $x \in Var$ is a term,
- every constant $c \in \mathbf{s}$ is a term,
- if $t_1, \dots, t_n$ are terms and $f \in \mathbf{s}$ is an n-ary function symbol, then $f(t_1, \dots, t_n)$ is a term.

Syntax: formulas

The set $\mathcal{L}_{\mathbf{s}}$ of **Formulas** over a given predicate signature **s** is defined inductively as follows:

- $\bot$ and $\top$ are formulas,
- if $t_1, \ldots, t_n$ are terms and $P \in \mathbf{s}$ is an n-ary predicate, then $P(t_1, \ldots, t_n)$ is a formula (called **atomic formula**),
- if φ, ψ are formulas, then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi$ are formulas,
- if $\varphi(x)$ is a formula and x a variable, then $\forall x \varphi(x)$ and $\exists x \varphi(x)$ are formulas.

A variable that does not fall within the scope of a quantifier is said to be a **free variable**, otherwise, it is said to be **bound**. A formula that has no free variable is a **closed formula**.

Fragments

Interesting **fragments** of the set of first order formulas $\mathcal{L}$ are the following:

- $\ddot{\mathcal{L}}$ is the fragment where **only binary and ternary predicates** are allowed,
- $\mathcal{L}^n$ is the set of formulas built up from a set of n **variables**.

We are indeed interested in the sets $\ddot{\mathcal{L}}^{\mathbf{2}}$ and $\ddot{\mathcal{L}}^3$.

Semantics: structures and assignments

A **first order structure M** for a given signature **s**, is a structure $\mathbf{M}=(M,(P^{\mathbf{M}})_{P\in\mathbf{s}},(f^{\mathbf{M}})_{f\in\mathbf{s}},(c^{\mathbf{M}})_{c\in\mathbf{s}})$, where:

- M is a non-empty set, called **domain**,
- for each predicate symbol $P \in \mathbf{s}$ of arity n, $P^{\mathbf{M}}$ is an n-ary relation on M,
- for each function symbol $f \in \mathbf{s}$ of arity n, $f^{\mathbf{M}}$ is an n-ary function on M and
- for each constant symbol $c \in \mathbf{s}$, $c^{\mathbf{M}}$ is an element of M.

An **assignment** α is a mapping $\alpha\colon \mathit{Var} \longrightarrow M$. Each assignment extends univocally to an assignment on terms satisfying, for every terms $t_1,\ldots,t_n$ and each n-ary function $f \in \mathbf{s}$, that

$$\alpha(f(t_1,\ldots,t_n)) = f^{\mathbf{M}}(\alpha(t_1),\ldots,\alpha(t_n)).$$

To denote that assignment α assigns objects $v_1,\ldots,v_n$ to variables $x_1,\ldots,x_n$, we will write $\alpha([v_1/x_1],\ldots,[v_n/x_n])$.

Semantics: models

Given a structure **M** assignment α and a formula φ, we say that **M** and α **satisfy** φ (in symbols $\mathbf{M}, \alpha \vDash \varphi$) if:

- if $\varphi = P(t_1, \ldots, t_n)$ then
$$P^{\mathbf{M}}(\alpha(t_1), \ldots, \alpha(t_n))$$
- if $\varphi = \psi \wedge \chi$, then
$$\mathbf{M}, \alpha \vDash \psi \qquad \text{and} \qquad \mathbf{M}, \alpha \vDash \chi$$
- if $\varphi = \neg\psi$, then
$$\mathbf{M}, \alpha \nvDash \psi$$
- if $\varphi = \exists x_1, \ldots, \exists x_n \varphi(x_1, \ldots, x_n)$, then exists $v_1, \ldots, v_n \in M$ such that
$$\mathbf{M}, \alpha \vDash \varphi(v_1, \ldots, v_n)$$

Logic

- We say that a formula φ is **satisfiable** if there exists a structure $\mathbf{M}$ and an assignment α such that

$$\mathbf{M}, \alpha \models \varphi.$$

- We say that a formula φ is a **tautology** if for every structure $\mathbf{M}$ and an assignment α it holds that

$$\mathbf{M}, \alpha \models \varphi.$$

- We say that a formula φ is **entailed** by a set of formulas Γ if for every structure $\mathbf{M}$ and an assignment α it holds that

$$\text{if } \mathbf{M}, \alpha \models \psi, \text{ for every formula } \psi \in \Gamma, \text{ then } \mathbf{M}, \alpha \models \varphi.$$

Translating Description Logic into first order logic

Translation of the signatures

Given a **description signature D** $= \langle N_I, N_C, N_R \rangle$, we define the **first order signature** $\mathbf{s_D} = N_I \cup N_C \cup N_R$, where

- N_I is the set of **constant** symbols,
- $N_C \cup N_R$ is the set of **unary and binary predicate** symbols.

For every concept name $A \in N_C$, every role name $R \in N_R$ and every $x, y \in Var$, we define the translations of **concept and role names**, respectively, into the set of atomic first order formulas in the following way:

$$\tau^x(A) := A(x)$$

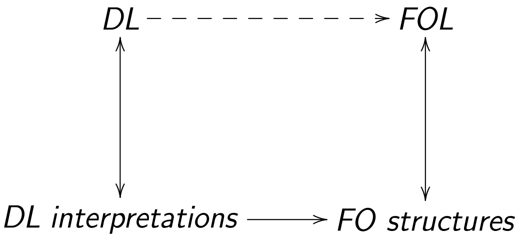
$$\tau^{x,y}(R) := R(x, y).$$

Translation of complex concepts in $\mathcal{ALCO}$

This translation can be inductively extended over the set of complex concept in $\mathcal{ALCO}$ in the following way:

$$
\begin{aligned}
\tau^x(\neg C) &:= \neg\tau^x(C) \\
\tau^x(C \sqcap D) &:= \tau^x(C) \wedge \tau^x(D) \\
\tau^x(C \sqcup D) &:= \tau^x(C) \vee \tau^x(D) \\
\tau^x(\forall R.C) &:= \forall y(\neg\tau^{x,y}(R) \vee \tau^y(C)) \\
\tau^x(\exists R.C) &:= \exists y(\tau^{x,y}(R) \wedge \tau^y(C)) \\
\tau^x\{a_1, \ldots, a_n\} &:= \{a_1, \ldots, a_n\}(x)
\end{aligned}
$$

Soundness of the translation



Inclusion in $\ddot{\mathcal{L}}^2$

$\mathcal{ALC}$ concepts can be expressed by means of $\ddot{\mathcal{L}}^2$ formulas. Indeed, there are needed just **two variables**.

- In the case of **nested quantifiers**, e.g.

$$\forall R.\exists R.\forall R.A$$

  we have that the translation is

$$\forall y(R(x,y) \rightarrow \exists x(R(y,x) \wedge \forall y(R(x,y) \rightarrow A(y))))$$

  and, since the inner variable "y" is **closed**, when a value of the outer quantifier "$\forall$" has to be calculated, this variable **falls outside its scope**.

- In case of **conjugated quantified concepts**, e.g.

$$(\forall R.A) \sqcap (\exists R.B)$$

we have that the translation is

$$(\forall y)(R(x,y) \rightarrow A(y)) \wedge (\exists y)(R(x,y) \wedge B(y))$$

where each appearance of variable "y" is closed **inside the scope of a different quantifier** and, for this reason, it does not fall inside the scope of the other quantifier.

Translation of axioms

- A **concept inclusion axiom** $C \sqsubseteq D$ can be translated in the following form:

$$\forall x(\tau^x(C) \rightarrow \tau^x(D))$$

- A **concept assertion axiom** $C(a)$ can be translated in the following form:

$$\tau^x(C)[a/x]$$

- A **role assertion axiom** $R(a, b)$ can be translated in the following form:

$$\tau^{x,y}(R)[a/x, b/y]$$

Translation of the reasoning tasks

- Since every reasoning task is reducible to **knowledge base consistency**, it is enough to translate this task.

- A **TBox** $\mathcal{T} = \{C_i \sqsubseteq D_i \colon 0 \leq i \leq n\}$ is satisfiable iff the formula

$$\forall x \bigwedge_{i=0}^{n} \tau^x(C_i) \rightarrow \tau^x(D_i)$$

  is satisfiable.

- An **ABox** $\mathcal{A} = \{C_j(a_i) \colon \langle i,j \rangle \in I\} \cup \{R_j(a_i, b_k) \colon \langle i,j,k \rangle \in J\}$ is satisfiable iff the formula

$$\bigwedge_{\langle i,j \rangle \in I} \tau^x(C_j)[a_i/x] \;\wedge\; \bigwedge_{\langle i,j,k \rangle \in J} \tau^{x,y}(R_j)[a_i/x, b_k/y]$$

  is satisfiable.

Translation of different role constructors

The translation of roles in the language $\mathcal{ALCROI}$ extends the one for $\mathcal{ALCO}$ in the following way:

$$
\begin{aligned}
\tau^{x,y}(\neg R) &:= \neg\tau^{x,y}(R) \\
\tau^{x,y}(R \sqcap S) &:= \tau^{x,y}(R) \land \tau^{x,y}(S) \\
\tau^{x,y}(R \sqcup S) &:= \tau^{x,y}(R) \lor \tau^{x,y}(S) \\
\tau^{x,y}(R^{-}) &:= \tau^{y,x}(R)
\end{aligned}
$$

Properties of the translation of complex roles

- As for $\mathcal{ALCO}$, also the **soundness** of the translation for complex roles in $\mathcal{ALCROI}$ is proved by means on a translation **between the respective semantics**.
- Again, it is easy to prove that **only** $\ddot{\mathcal{L}}^2$ **formulas** can be obtained.
- A **role inclusion axiom** $R \sqsubseteq P$ can be translated in the following form:

$$\forall x \forall y (\tau^{x,y}(R) \rightarrow \tau^{x,y}(P))$$

Translation of role composition

- The translation of roles in the language $\mathcal{ALCROI}(\circ)$ extends the one for $\mathcal{ALCROI}$ in the following way:

$$\tau^{x,y}(R \circ S) \;:=\; \exists z(\exists y(y = z \wedge \tau^{x,y}(R)) \wedge \exists x(x = z \wedge \tau^{x,y}(S)))$$

- For **longer chains** of composed roles, the composition is defined as a binary operation:

$$\tau^{x,y}(R_1 \circ R_2 \circ R_3 \ldots \circ R_{n-1} \circ R_n) =$$

$$\tau^{x,y}(\tau^{x,y}(\ldots \tau^{x,y}(\tau^{x,y}(R_1 \circ R_2) \circ R_3) \ldots \circ R_{n-1}) \circ R_n)$$

- Since the inner variable "z" is **closed**, when a value of the outer quantifier "$\exists$" has to be calculated, this variable **falls outside its scope**.

- Hence, the translation of $\mathcal{ALCROI}(\circ)$ can be expressed by means of $\ddot{\mathcal{L}}^3$ **formulas**.

Translation of cardinality restriction

This translation of roles in the language $\mathcal{ALCON}$ and $\mathcal{ALCOQ}$ can be made as an extension of the one for $\mathcal{ALCO}$ in two ways:

- by allowing an **unbounded number of variables**, so translating:

$$\tau^{x_1,\dots,x_n}(\geq nR) \quad := \quad \exists x_1 \dots \exists x_n(R(x,x_1),\dots,R(x,x_n))$$

- by allowing an **bounded quantifiers**, so obtaining the F.O. fragment $\ddot{\mathcal{C}}^2$ and translating:

$$\tau^{x,y}(\geq nR) \quad := \quad \exists_{\geq n} y(R(x,y))$$

But both ways are essentially equivalent and go **beyond** $\ddot{\mathcal{C}}^2$.

Summary

DL	FOL
$\mathcal{ALCO}$	$\ddot{\mathcal{L}}^2$
$\mathcal{ALCROI}$	$\ddot{\mathcal{L}}^2$
$\mathcal{ALCROI}(\circ)$	$\ddot{\mathcal{L}}^3$
$\mathcal{ALCON}$	$\ddot{\mathcal{C}}^2$
$\mathcal{ALCOQ}$	$\ddot{\mathcal{C}}^2$

Translating first order logic into Description Logic

Translating FOL into DL

- In general it is **not possible** to obtain a syntactical translation of the full first order logic into any DL language.
- Indeed, there are some **FOL formulas** that cannot be defined by any DL language:
  - formulas with **predicates with arity** ≥ 2 cannot be used in DL concepts, since there are just unary and binary predicates,
  - formulas with **more than one free variable** cannot be expressed as DL concepts, since these express just unary relations in the domain set,
  - formulas with **global quantification** cannot be expressed as DL concepts, since these only quantify on the successors of a given node.

Translating fragments into DL

In A. Borgida, *On the relative expressiveness of Description Logics and predicate logics* it is proved that the fragments $\ddot{\mathcal{L}}^2$ and $\ddot{\mathcal{L}}^3$ can be indeed translated into $\mathcal{ALCROI}$ and $\mathcal{ALCROI}(\circ)$ **with some modifications**:

- the following two **role constructors are introduced**:

identity	*id*	$\{\langle v, v\rangle : v \in \Delta^{\mathcal{I}}\}$
cross-product	$C \times D$	$\{\langle v, w\rangle : v \in C^{\mathcal{I}}, w \in D^{\mathcal{I}}\}$

- **Roles are treated as concept**, in the sense that they can appear outside complex concepts or axioms.

Now:

- the restriction of the signature to $\ddot{\mathcal{L}}$ allows to restrict to formulas with just **binary and unary predicates**;
- the restriction of the language to $\mathcal{L}^2$ or $\mathcal{L}^3$ allows to restrict to formulas with just **two or three variables**;
- the modifications to the DL languages above provided allow to **translate formulas with up to three free variables**;
- the global quantification can be treated through the use of a **universal role**, which, in languages with role constructors, can be obtained as $R \sqcup \neg R$.