Description Logics: a Nice Family of Logics
— Relation with Other Logics —

Uli Sattler[1] *Thomas Schneider*[2]

[1]School of Computer Science, University of Manchester, UK

[2]Department of Computer Science, University of Bremen, Germany

ESSLLI, 16 August 2016

Previous and next steps

So far: syntax, semantics, and basics of the DL $\mathcal{ALC}$:

- where they come from
- **Syntax**: concepts, axioms, assertions, TBox, ABox, ontology
- **Semantics**: interpretations, models
- **Reasoning problems**: entailment, satisfiability, consistency, ... and relationships between reasoning problems

Next: relationships between

- Description Logic
- Modal Logic
- First Order Logic

A brief recap of the history of DLs

Description Logics were

- developed as logical formalisation of semantic networks in the late 1980s
- discovered to have close relationships with FOL, ML in the early 1990s
- investigated widely in the last 25+ years:
  - trade-off between expressive power and computational complexity of reasoning
  - model theory
  - ...
- used as the logical basis of the Web Ontology Language, OWL

Relationship with first-order logic (FOL)

Not hard to see:

If we view concept names A as unary predicates and roles r as binary predicates, then

- each interpretation $\mathcal{I}$ can be seen as an FOL structure;
- each $\mathcal{ALC}$ concept C can be translated into an FOL formula $\boldsymbol{t_x(C)}$ with one free variable x such that:

$$a \in C^{\mathcal{I}} \quad \text{iff} \quad \mathcal{I} \models t_x(C)[x/a]$$

Translation: see next slide

Translation of $\mathcal{ALC}$ concepts into FOL formulas

$$
\begin{aligned}
t_x(A) &= A(x) & t_y(A) &= A(y)\\
t_x(\neg C) &= \neg t_x(C) & t_y(\neg C) &= \ldots\\
t_x(C \sqcap D) &= t_x(C) \wedge t_x(D) & t_y(C \sqcap D) &= \ldots\\
t_x(C \sqcup D) &= \ldots & t_y(C \sqcup D) &= \ldots\\
t_x(\exists r.C) &= \exists y.r(x,y) \wedge t_y(C) & t_y(\exists r.C) &= \ldots\\
t_x(\forall r.C) &= \ldots & t_y(\forall r.C) &= \ldots
\end{aligned}
$$

Exercise 1:

- Fill in the blanks
- Why are $t_x(C)$, $t_y(C)$ formulas in one free variable?
- Translate the concept $\neg A \sqcup \exists r.\forall s.B$ into an FOL formula.

Translation of ontologies into FOL formulas

Translate an ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ using $t()$ as follows:

$$
\begin{aligned}
t(\mathcal{O}) &= t(\mathcal{T}) \cup t(\mathcal{A})\\
t(\mathcal{T}) &= \{\forall x.t_x(C) \rightarrow t_x(D) \mid C \sqsubseteq D \in \mathcal{T}\}\\
t(\mathcal{A}) &= \{t_x(C)[x/a] \mid a : C \in \mathcal{A}\} \cup \{r(a,b) \mid (a,b) : r \in \mathcal{A}\}
\end{aligned}
$$

Consequence:

Theorem

1. a is an instance of C in $\mathcal{I}$ iff $\mathcal{I} \models t_x(C)[x/a]$.
2. C is satisfiable iff $t_x(C)$ is satisfiable.
3. C is satisfiable w.r.t. $\mathcal{O}$ iff $\{t_x(C)[x/a]\} \cup t(\mathcal{O})$ is satisfiable.
4. C is subsumed by D iff $\forall x.(t_x(C) \rightarrow t_x(D))$ is valid.
5. $\mathcal{O} \models C \sqsubseteq D$ iff $t(\mathcal{O}) \models \forall x.(t_x(C) \rightarrow t_x(D))$.

Locating DLs within FOL

Observations. $t_x(C)$ uses

- only two variables

  $\Rightarrow$ $\mathcal{ALC}$ is a fragment of the **2-variable fragment of FOL**, which is known to be decidable.

- only guarded quantification

  $\Rightarrow$ $\mathcal{ALC}$ is a fragment of the **guarded fragment of FOL**, which is known to be decidable.

Relationship with modal logic (ML)

Easy case: only one role is used, e.g.:

DL concept C	$\rightsquigarrow$	ML formula $\varphi(C)$
$A \sqcap \exists r.(A \sqcap B)$		$A \wedge \Diamond(A \wedge B)$
$A \sqcap \forall r.(A \sqcap B)$		$A \wedge \Box(A \wedge B)$
$A \sqcap \exists r.A \sqcap \forall r.B$		$A \wedge \Diamond A \wedge \Box B$
$A \sqcap \exists r.A \sqcap \forall r.\neg A$		$A \wedge \Diamond A \wedge \Box\neg A$

General case: switch to **multi-modal logic (MML)**, e.g.:

$$A \sqcap \exists r.A \sqcap \forall s.(\neg A \sqcap \exists t.B) \rightsquigarrow A \wedge \langle r\rangle A \wedge [s](\neg A \wedge \langle t\rangle B)$$

MML extends the ML ...

- syntax to parameterised boxes & diamonds, and
- semantics to several accessibility relations R_s, e.g.,

  $\mathcal{M}, w \models [s]\varphi$ if, for all $v \in W$, $(w,v) \in R_s$ implies $\mathcal{M}, v \models \varphi$

Relationship with ML: ontologies

In ML, we are mainly concerned with a single formula.
There is no equivalent to TBoxes or ABoxes, but:

- **TBox:**

  if we have the **universal modality** u, we can translate

  $$C \sqsubseteq D \quad \text{into} \quad [u](\neg\varphi(C) \vee \varphi(D))$$

- **ABox:**

  if we have **nominals,** we can translate

  $$a : C \quad \text{into} \quad @_a(\varphi(C))$$
  $$(a, b) : r \quad \text{into} \quad @_a\langle r\rangle b$$

Exercise 2:
Translate the TBox $\{\neg A \sqsubseteq \exists r.\forall s.B\}$ into an ML formula.

Relationship with ML: harvest

Without TBoxes, we can use the known

- algorithms for modal logic (MLAs) to **decide** satisfiability and subsumption in $\mathcal{ALC}$
- soundness & completeness proof of the MLA to show that $\mathcal{ALC}$ has the
  - **finite model property:**
    C is sat. iff C is sat. in an interpretation with finite domain.
  - **tree model property:**
    C is sat. iff C is sat. in a tree-shaped interpretation.
  - **finite tree model property:**
    C is sat. iff C is sat. in a finite tree-shaped interpretation.

With TBoxes, dedicated techniques are required
to decide sat. & subsumption efficiently in practice $\rightsquigarrow$ **over to Uli!**

References

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider (editors).
The Description Logic Handbook:
Theory, Implementation and Applications.
2nd edition, Cambridge University Press, 2007.
ISBN 978-0521876254.
Chapter 4.