Reasoning and Planning
Unit 6. Terminological Reasoning

Pedro Cabalar

Dept. Computer Science
University of Corunna, SPAIN

November 18, 2022

Description Logic

- Domain knowledge uses many concepts. E.g. a piston is a mechanical piece and is part of the engine that is a device ...

☞ Represent concept definitions for a given domain = Terminological Knowledge

- In the past known as terminological systems or concept languages
- Nowadays, main use: description of Ontologies for the Semantic Web (OWL-DL, OWL-Lite).
- Description Logic (DL) uses basic operators (similar to modal logic) that can be translated to predicate calculus.

Description Logic

- Syntax: we start from
  1. A set of atoms called concepts. They represent classes or sets of objects. E.g.: *Person*, *Mammal*, *Vehicle*, *Blue*, ...
  2. A set of roles that represent binary relations among objects. E.g.: *likes*, *owns*, *travels_by*, *has_child* ...
  3. A set of constructors to define new concepts recursively. E.g.: blue or red vehicles = $Vehicle \sqcap (Blue \sqcup Red)$.
     E.g.: a father is a $Man \sqcap \exists has\_child$
- There exists a whole family of Description Logics depending on the constructors we allow

Description Logic: $\mathcal{FL}^-$

- The simplest DL is $\mathcal{FL}^-$. Syntax:

$$C \quad ::= \quad A \mid C \sqcap D \mid \forall R.C \mid \exists R \mid \top \mid \bot$$

  where A=atomic concept, C, D=concepts and R=role.

- Alternative syntax:

$$C \quad ::= \quad A \mid (\texttt{:and}\ \ C\ D) \mid (\texttt{:all}\ \ R\ C) \mid (\texttt{:some}\ \ R)$$

- Quantifiers: $\forall has\_child.Female$ are those living beings whose offsprings are all female, whereas $\exists has\_child$ are those that have, at list, some child. We can write it as $\exists has\_child.\top$ too

Description Logic: $\mathcal{FL}^-$

Definition (Interpretation)

An interpretation $\mathcal{I}$ is a pair $\langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ where

- $\Delta^\mathcal{I}$ is a non-empty set called the domain
- $\cdot^\mathcal{I}$ is a function that maps:
  1. Each concept to a subset of $\Delta^\mathcal{I}$.
  2. Each role to a subset of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$.

Description Logic: $\mathcal{FL}^-$

- We extend $\cdot^{\mathcal{I}}$ for evaluation of non-atomic concepts:

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(\exists R)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \; : \; \exists y.(x,y) \in R^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \; : \; \forall y.(x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\} \\
(\top)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
(\bot)^{\mathcal{I}} &= \emptyset
\end{aligned}
$$

- Examples: which is the meaning of the following expressions?

$$
\begin{array}{l}
Adult \sqcap Male \\
\forall has\_child.(Adult \sqcap Male) \\
\exists has\_child \sqcap \forall has\_child.(\exists has\_child \sqcap Adult)
\end{array}
$$

Description Logic: $\mathcal{ALC}$

- A second example: $\mathcal{ALC}$ is more expressive than $\mathcal{FL}^-$. It allows:

$$C \quad ::= \quad A \mid C \sqcap D \mid \forall R.C \mid \exists R \mid \neg C$$

con $(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$

- A more limited variant $\mathcal{AL}$ replaces $\neg C$ by $\neg A$ (only atomic concepts can be negated).

Description Logics: most common constructors

Constructor	Sintaxis	Semántica
Concept assertion	$o : C$	$o^{\mathcal{I}} \in C^{\mathcal{I}}$
Role assertion	$(o_1, o_2) : R$	$(o_1^{\mathcal{I}}, o_2^{\mathcal{I}}) \in R^{\mathcal{I}}$
Intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Union	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Existential	$\exists R.C$	$\{x \ : \ \exists y.(x,y) \in R^{\mathcal{I}} \& y \in C^{\mathcal{I}}\}$
Universal	$\forall R.C$	$\{x \ : \ \forall y.(x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$
Cardinality	$(\geq n\ R)$	$\{x \ : \ \lvert\{y.(x,y) \in R^{\mathcal{I}}\}\rvert \geq n\}$
	$(\leq n\ R)$	$\{x \ : \ \lvert\{y.(x,y) \in R^{\mathcal{I}}\}\rvert \leq n\}$
Inverse	R^{-}	$\{(x,y) \ : \ (y,x) \in R^{\mathcal{I}}\}$
Transitive	R^{*}	$(R^{\mathcal{I}})^{*}$
Enumeration (one-of)	$\{o_1, \ldots, o_n\}$	$\{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\}$

...

Description Logics: naming

- To name each combination, we use the initials

  $\mathcal{F}$ functional property $(\leq 1\ R)$
  $\mathcal{E}$ general existential $\exists R.C$
  $\mathcal{U}$ concept union $C \sqcup D$
  $\mathcal{C}$ concept negation $\neg C$
  $\mathcal{S}$ = $\mathcal{ALC}$ plus transitive roles R^*
  $\mathcal{H}$ role hierarchy
  $\mathcal{R}$ reflexive, irreflexive and disjunct roles
  $\mathcal{I}$ inverse role R^-
  $\mathcal{O}$ enumeration (one-of) $\{o_1, \ldots, o_n\}$
  $\mathcal{N}$ cardinality constraints $(\geq n\ R)$, $(\leq n\ R)$
  $\mathcal{Q}$ qualified cardinality constraints $(\geq n\ R.C)$, $(\leq n\ R.C)$
  $(\mathcal{D})$ data types (integer, string, etc)

- OWL-DL corresponds to $\mathcal{SHOIN}^{(\mathcal{D})}$ whereas OWL-Lite is based on $\mathcal{SHIF}^{(\mathcal{D})}$.

Description Logic

- A (terminological) Knowledge Base consists of two sets of expressions TBox and ABox.
- TBox: contains general terminological declarations. Two types:
  1. A concept defition $A \equiv C$. Examples:

$$\begin{aligned} Woman &\equiv Person \sqcap Female \\ Mother &\equiv Woman \sqcap \exists has\_child \end{aligned}$$

     (acyclicity is usually required)
  2. An inclusion axiom $C_1 \sqsubseteq C_2$. Example

$$\exists has\_child.Person \sqsubseteq Person$$

     They impose constraints on our model

Description Logic

- ABox: contains assertions about specific element and relation instances in the domain. Two types:
  1. Concept assertion $o : C$. Example:

$$Moby\_Dick : Whale$$
$$Mary : Female \sqcap \exists has\_child$$

  2. Role assertions $(o_1, o_2) : R$. Example:

$$(Mary, Jesus) : has\_child$$

Description Logic: reasoning

- Typical reasoning problems to solve. Given a Knowledge Base:
  1. *Subsumption*. Check whether a concept is more general than another $C \sqsubseteq D$
  2. Equivalence. Check whether two concepts are equivalent $C \equiv D$
  3. Consistency. Check whether a concept has at least some model $C \equiv \bot$
  4. Belonging. Check whether an individual is member of a concept $o : C$
- All these problems can be reduced to consistency. Example: $C \sqsubseteq D$ can be checked as the consistency test $C \sqcap \neg D \equiv \bot$.

Description Logic: reasoning

- There exist efficient methods to solve these reasoning problems
- Normally, more expressive language variant $\Rightarrow$ more complex associated reasoning
- Example: subsumption in $\mathcal{FL}^-$ is decidable in time complexity P.
- See Description Logic Complexity Navigator
  http://www.cs.man.ac.uk/~ezolin/dl/

Description Logic: translation to First-Order Logic

- Most Description Logics are reducible to decidable fragments of Firt-Order Logic
- Each concept C becomes a unary predicate $C(x)$; each role R a binary predicate $R(x, y)$.
- We can use First-Order Logic with variables:

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

- In a TBox, we translate $C \equiv D$ into $\forall x.t_x(C) \leftrightarrow t_x(D)$.