G53KRR 2013-2014 handout on descripton logic

OWL Web Ontology Language - W3C standard, extends most description logics and has slightly different terminology (based on RDF rather than description logic semantics. OWL DL based on DL).

Reading:

The Description Logic Handbook. Franz Baader, Diego Calvanese, Deborah L.McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. Cambridge University Press, 2003. ISBN 0-521-78176-0.

A good on-line course: `http://www.inf.unibz.it/%7Efranconi/dl/course/`

Basic idea description logics talk about relationships between *concepts* (noun phrases). There are many different description logics:

Description logic ALC :

Logical symbols (apart from brackets etc.):

- concept-forming operators: $\forall$, $\exists$, $\sqcup$, $\sqcap$, $\neg$
- connectives: $\sqsubseteq$, $\doteq$

Non-logical symbols:

- Atomic concepts: $Person$, $Thing$,.... Correspond to unary predicates in FOL.
- Roles: Age, $Employer$, $Child$, ... Correspond to binary predicates in FOL.
- Constants: $john, mary, roomA7, \ldots$. Correspond to constants (0-ary functional symbols) in FOL.

Concepts:

- atomic concept is a concept
- if R is a role and C is a concept, then $\forall R.C$ is a concept (e.g. $\forall Child.Girl$ describes someone all of whose children are girls)
- if R is a role and C is a concept, then $\exists R.C$ is a concept (e.g. $\exists .Child.Girl$ describes someone who has a daughter)
- if C is a concept then $\neg C$ is a concept
- if C_1 and C_2 are concepts then $C_1 \sqcap C_2$ is a concept
- if C_1 and C_2 are concepts then $C_1 \sqcup C_2$ is a concept

Sentences:

- if C_1 and C_2 are concepts then $C_1 \sqsubseteq C_2$ is a sentence (all C_1s are C_2s, C_1 is *subsumed* by C_2)
- if C_1 and C_2 are concepts then $C_1 \doteq C_2$ is a sentence (C_1 is equivalent to C_2)
- if a is a constant and C a concept then $C(a)$ is a sentence (the individual denoted by a satisfies the description expressed by C)
- if a, b are constants and R a role then $R(a, b)$ is a sentence (the individuals denoted by a and b are connected by the role R)

A description logic knowledge base is a set of sentences.

TBox and ABox A description logic knowledge base is usually split into terminological part or TBox which describes general relationships between concepts, e.g. $Surgeon \sqsubseteq Doctor$, and assertions about individuals or ABox (e.g. $Doctor(mary)$).

Interpretations for description logic same as for FOL: a set of individuals D and an interpretation mapping I such that

- for a constant a, $I(a) \in D$
- for an atomic concept A, $I(A) \subseteq D$
- for a role R, $I(R) \subseteq D \times D$
- $I(\forall R.C) = \{x \in D : \text{for any } y, \text{ if } (x, y) \in I(R), \text{ then } y \in I(C)\}$. Same as $\forall y(R(x, y) \supset C(y))$
- $I(\exists R.C) = \{x \in D : \text{there is a } y \text{ such that } (x, y) \in I(R) \text{ and } y \in I(C)\}$. Same as $\exists y(R(x, y) \wedge C(y))$
- $I(\neg C) = D \setminus I(C)$
- $I(C_1 \sqcap C_2) = I(C_1) \cap I(C_2)$. Same as $C_1(x) \wedge C_2(x)$
- $I(C_1 \sqcup C_2) = I(C_1) \cup I(C_2)$. Same as $C_1(x) \vee C_2(x)$

Finally, for sentences:

- $(D, I) \models C(a)$ iff $I(a) \in I(C)$. Same as $C(a)$
- $(D, I) \models R(a, b)$ iff $(I(a), I(b)) \in I(R)$. Same as $R(a, b)$
- $(D, I) \models C_1 \sqsubseteq C_2$ iff $I(C_1) \subseteq I(C_2)$. Same as $\forall x(C_1(x) \supset C_2(x))$
- $(D, I) \models C_1 \doteq C_2$ iff $I(C_1) = I(C_2)$. Same as $\forall x(C_1(x) \equiv C_2(x))$

Reasoning Entailment is defined exactly like in FOL: a set of sentences Γ entails a sentence ϕ (in symbols $\Gamma \models \phi$) if and only if ϕ is true in every interpretation where all of the sentences in Γ are true.

ALC is a proper fragment of first order logic. Reasoning in ALC it is decidable (it is decidable whether a sentence is satisfiable, or whether a finite set of sentences entails another sentence; however algorithms for checking this take exponential time).

Example of a description logic where reasoning is very efficient: EL only has $\sqcap$ and $\exists.R$ as concept constructors. Reasoning not just decidable, but very efficient (polynomial time algorithm for checking subsumption of concepts).

Other features used to define more expressive description logics: functional roles (for example, to say that only one object can be connected by an Age role), cardinality restrictions on the number of objects connected by a role, ability to say that roles are transitive, reflexive, express inclusion relation between roles. Some very expressive description logics are undecidable.