Description Logics Extending $\mathcal{ALC}$
Extending \textit{ALC}

We discuss the extension of \textit{ALC} by

- qualified number restrictions;
- inverse roles;
- transitive roles;
- roles inclusions;
- nominals.
Extending $\mathcal{ALC}$ by Qualified Number Restrictions

**Qualified number restrictions**: if $C$ is a concept, $r$ a role, and $n$ a number, then

$$(\leq n \ r. C), \ (\geq n \ r. C)$$

are concepts. If $S$ is a set, then we denote by $|S|$ the number of its elements. The interpretation of qualified number restrictions is given by

- $(\leq n \ r. C)^I = \{ x \in \Delta^I \mid |\{ y \in \Delta^I \mid (x, y) \in r^I \text{ and } y \in C^I \}| \leq n \}$
- $(\geq n \ r. C)^I = \{ x \in \Delta^I \mid |\{ y \in \Delta^I \mid (x, y) \in r^I \text{ and } y \in C^I \}| \geq n \}$

**Examples**

- $(\geq 3 \ \text{hasChild. Male})$ is the class of all objects having at least three children who are male.

- $(\leq 2 \ \text{hasChild. Male})$ is the class of all objects having at most two children who are male.
We have seen unqualified number restrictions in DL-Lite. Recall that unqualified number restrictions are of the form

\[ \leq n \, r \, T \]

and do not admit qualifications using an arbitrary concept $C'$. DL-Lite does not admit such qualifications because terminological reasoning would become ExpTime-hard.
Extending $\mathcal{ALC}$ by inverse roles

**Inverse roles**: If $r$ is a role name, then $r^-$ is a role, called the inverse of $r$. The interpretation of inverse roles is given by

\[
(r^-)^I = \{(y, x) \in \Delta^I \times \Delta^I \mid (x, y) \in r^I\}.
\]

$r^-$ can occur in all places in which the role name $r$ can occur.

**Examples**

- $\exists \text{has\_child}^- . \text{Gardener}$ is the class of all objects having a parent who is a gardener.

- $(\geq 3 \text{parent}^- . \text{Gardener})$ is the class of all objects having at least three children who are gardeners.

We have seen inverse roles in DL-Lite. There are no inverse roles in $\mathcal{EL}$. In fact, adding inverse roles to $\mathcal{EL}$ would make reasoning ExpTime-hard.
Extending $\mathcal{ALC}$ by transitive roles and role hierarchies

**Transitive roles**: One can add $\text{transitive}(r)$ to a TBox to state that the relation $r$ is transitive. Thus,

- $\mathcal{I} \models \text{transitive}(r)$ if, and only if, $r^\mathcal{I}$ is transitive, i.e., for all $x, y, z \in \Delta^\mathcal{I}$ such that $(x, y) \in r^\mathcal{I}$ and $(y, z) \in r^\mathcal{I}$ we have $(x, z) \in r^\mathcal{I}$.

**Examples**

- The role “is part of” is often regarded as transitive.

**Role hierarchies**: one can add a role inclusion $r \sqsubseteq s$ to a TBox to state that $r$ is included in $s$. Thus,

- $\mathcal{I} \models r \sqsubseteq s$ iff $r^\mathcal{I} \subseteq s^\mathcal{I}$.

**Example**:

- hasSon $\sqsubseteq$ hasChild
Extending $\mathcal{ALC}$ by Nominals

Sometimes we want to use concepts/classes consisting of exactly one object or a finite set of objects. To enable the construction of such concepts, $\mathcal{ALC}$ has been extended by nominals.

**Nominals:** We use $a, b$, etc. to denote individual names. Individual names denote elements of the domain of interpretations. They are names for individual objects (not for classes or relations). Thus, we extend interpretations $I$ to interpret individual names by setting $a^I \in \Delta^I, b^I \in \Delta^I$, etc.

For every individual name $a$, we call $\{a\}$ a nominal. For individual names $a_1, \ldots, a_n$, we call $\{a_1, \ldots, a_n\}$ a nominal set.

In every interpretation $I$:

- $\{a\}^I = \{a^I\}$;
- $\{a_1, \ldots, a_n\}^I = \{a_1^I, \ldots, a_n^I\}$. 

Ontology Languages
Extending $\mathcal{ALC}$

In $\mathcal{ACC}$ extended with nominals we can use the expressions $\{a\}$ and $\{a_1, \ldots, a_n\}$ as concepts.

Examples:

- $\exists \text{citizen\_of}. \{\text{France}\}$ (citizens of France).

- $\exists \text{citizen\_of}. \{\text{France, Ireland}\}$ (citizens of France or Ireland).

- $\exists \text{has\_colour}. \{\text{Green}\}$ (all green objects).

- $\exists \text{student\_of}. \{\text{Liverpool\_University}\}$ (students of Liverpool University).

- One can also define the concept $\text{Colour}$ by giving a list of all colours:

  $$\text{Colour} \equiv \{\text{red, yellow, \ldots, green}\}$$

  and give a value restriction for the role $\text{has\_colour}$ by

  $$\top \sqsubseteq \forall \text{has\_colour}. \text{Colour}.$$
The expressive Description Logics $\text{SHOIQ}$

The extension of $\text{ALC}$ with the constructors

- qualified number restrictions,
- inverse roles,
- role hierarchies,
- transitive roles,
- and nominals

is called $\text{SHOIQ}$. It is the underlying description logic of the Web Ontology Language OWL-DL we will discuss later. Standard reasoning systems (FACT, RACER, Pellet) for $\text{SHOIQ}$ are based on tableau procedures similar to the one discussed for $\text{ALC}$. Similar to $\text{ALC}$, terminological reasoning in $\text{SHOIQ}$ is decidable, but not tractable (it is ExpTime hard).