Handout: Description logic DL-Lite$_R$

Syntax of DL-Lite$_R$

In DL-Lite$_R$, TBox inclusions are of one of the following four forms

\[ B_1 \sqsubseteq B_2 \quad B_1 \sqsubseteq \neg B_2 \quad S_1 \sqsubseteq S_2 \quad S_1 \sqsubseteq \neg S_2 \]

where

\[ B := A \mid \exists S \quad S := r \mid r^- \]

with $A$ a concept name and $r$ a role name.

DL-Lite$_R$ ABoxes only contain assertions of the forms $A(c)$ and $r(c,d)$ with $A$ a concept name and $r$ a role name.

Queries

A conjunctive query (CQ) has the form $q(x_1, \ldots, x_k) = \exists y_1, \ldots, y_m \alpha_1 \land \ldots \land \alpha_n$ where each $\alpha_i$ is an atom of the form $A(t_1)$ or $r(t_1, t_2)$ or $t_1 = t_2$, and each $t_j$ is either a variable drawn from $x_1, \ldots, x_k, y_1, \ldots, y_m$ or an individual name. We call the free variables $x_1, \ldots, x_k$ the answer variables, and $y_1, \ldots, y_m$ the existential variables.

A first-order (FO) query is similarly built from concept, role, and equality atoms, but may use existential and universal quantification, conjunction, disjunction, and negation.

The certain answers to a query $q(x_1, \ldots, x_n)$ w.r.t. a DL KB $(T, A)$ are all tuples of individuals $(a_1, \ldots, a_n)$ drawn from $A$ such that $q(\bar{a})$ is satisfied in every model of $(T, A)$ (where $q(\bar{a})$ is obtained by replacing each $x_i$ by $a_i$). We denote the set of certain answers by $\text{cert}(q, (T, A))$.

Definition of query rewriting

Given an ABox $A$, we define the interpretation $\mathcal{I}_A$ as follows:

- $\Delta^{\mathcal{I}_A}$ contains the individuals in $A$
- $A^{\mathcal{I}_A} = \{a \mid A(a) \in A\}$
- $r^{\mathcal{I}_A} = \{(a, b) \mid r(a, b) \in A\}$
- $a^{\mathcal{I}_A} = a$ for every individual $a$ in $A$

A query $q'$ is called a rewriting of a CQ $q$ w.r.t. a TBox $T$ iff

\[ \bar{a} \in \text{cert}(q, (T, A)) \iff \mathcal{I}_A \models q'(\bar{a}) \]

for every ABox $A$. 
Query rewriting procedure

The rewriting ra(α, I) of atom α using applicable inclusion I is defined by:

- if α = A(x) and I = B ⊑ A, then ra(α, I) = B(x)
- if α = A(x) and I = ∃r ⊑ A, then ra(α, I) = r(x, .)
- if α = A(x) and I = ∃r− ⊑ A, then ra(α, I) = r(., x)
- if α = r(x, .) and I = A ⊑ ∃r, then ra(α, I) = A(x)
- if α = r(x, .) and I = ∃s ⊑ ∃r, then ra(α, I) = s(x, .)
- if α = r(x, .) and I = ∃s− ⊑ ∃r−, then ra(α, I) = s(., x)
- if α = A(x, y) and I = s ⊑ r or I = s− ⊑ r−, then ra(α, I) = s(x, y)
- if α = r(x, y) and I = s ⊑ r− or I = s− ⊑ r, then ra(α, I) = s(y, x)

Note that in the above, x and y can be variables, individuals, or the symbol .

We now give the PerfectRef query rewriting algorithm:

**Input:** TBox T, conjunctive query q₀ (with no ∃-vars in equality atoms)

**Output:** finite set of CQs (which may use special symbol ' .')

PR := {τ(q₀)}
repeat until PR' = PR

    PR' := PR
    for each q ∈ PR' that has not yet been considered do
        for each α ∈ q and I ∈ T do
            if ra(α, I) is defined
                PR := PR ∪ {q[α/ra(α, I)]}
            for each α, β ∈ q do
                if α and β unify
                    PR := PR ∪ {τ(merge(q, α, β))}

return PR

q[α/ra(α, I)]: replace atom α in q by ra(α, I)

τ: replaces existential variables that occur only once by .

merge: applies most general unifier of α and β, adds x = t if answer variable x was replaced by term t

Rewriting rewrite(q, T) is the query obtained by taking the disjunction of all queries in PerfectRef(q, T), with each . symbol replaced by a fresh existential variable.