6. Reasoning in Description Logics

Exercise 6.1 Let $\mathcal{T}$ be a TBox consisting of concept inclusions of the form $A_1 \sqsubseteq A_2$ and concept disjointness assertion of the form $A_1 \sqsubseteq \neg A_2$, for atomic concepts $A_1$ and $A_2$.

Describe an algorithm for checking concept satisfiability with respect to $\mathcal{T}$, i.e., whether for some concept $A$ it holds that $A$ is satisfiable with respect to $\mathcal{T}$.

What is the complexity of the algorithm?

Solution: Let $C$ be the set of atomic concepts appearing in $\mathcal{T}$. Construct a directed graph $G_\mathcal{T} = (N, E)$ as follows:

- the set of nodes is $N = C \cup \{\neg A \mid A \in C\}$;
- the set of directed edges is $E = \{A_1 \rightarrow A_2, \neg A_2 \rightarrow \neg A_1 \mid A_1 \sqsubseteq A_2 \in \mathcal{T}\} \cup \{A_1 \rightarrow \neg A_2, A_2 \rightarrow \neg A_1 \mid A_1 \sqsubseteq \neg A_2 \in \mathcal{T}\}$.

Then one can show that an atomic concept $A$ is unsatisfiable with respect to $\mathcal{T}$ if and only if there is a path from $A$ to $\neg A$. The algorithm for reachability checking can be done in linear time.

NOTE: the reachability checking problem is in NLOGSPACE.

Exercise 6.2 Consider TBoxes $\mathcal{T}$ consisting of axioms of the forms

\[
B_1 \sqsubseteq B_2, \quad \text{where} \quad B_1, B_2 := A \mid \exists P \mid \exists P^-, \\
R_1 \sqsubseteq R_2, \quad \text{where} \quad R_1, R_2 := P \mid P^-,
\]

where $A$ denotes an atomic concept, and $P$ an atomic role.

- Describe an algorithm for checking concept subsumption with respect to a given $\mathcal{T}$, i.e., whether for two concepts $B_1$ and $B_2$ it holds that $\mathcal{T} \models B_1 \sqsubseteq B_2$.

- Let $\mathcal{A}_0 = \{A_0(a)\}$, for some atomic concept $A_0$ and individual $a$, and let $\mathcal{T}$ be a(n arbitrary) TBox of the above form. Can we determine whether $(\mathcal{T}, \mathcal{A}_0)$ is satisfiable?

Solution: Let $C$ be the set of atomic concepts and $\mathcal{R}$ the set of atomic roles appearing in $\mathcal{T}$. For an atomic or inverse role $R$, we use $R^-$ to denote $P^-$ if $R$ is an atomic role $P$, and to denote $P$ if $R$ is an inverse role $P^-$.

Construct a directed graph $G_\mathcal{T} = (N, E)$ as follows:

- the set of nodes is $N = C \cup \{\exists P \mid P \in \mathcal{R}\} \cup \{\exists P^- \mid P \in \mathcal{R}\}$;
- the set of directed edges is $E = \{B_1 \rightarrow B_2 \mid B_1 \sqsubseteq B_2 \in \mathcal{T}\} \cup \{\exists R_1 \rightarrow \exists R_2 \mid R_1 \sqsubseteq R_2 \in \mathcal{T}\} \cup \{\exists R_1^- \rightarrow \exists R_2^- \mid R_1 \sqsubseteq R_2 \in \mathcal{T}\}$.

Then one can show that $\mathcal{T} \models B_1 \sqsubseteq B_2$ if and only if there is a path from $B_1$ to $B_2$ in $G_\mathcal{T}$.

The TBox $\mathcal{T}$ does not contain assertions involving negation. Hence, every knowledge base having $\mathcal{T}$ as TBox and an arbitrary ABox (including $\mathcal{A}_0$) is satisfiable.

Exercise 6.3 Show that concept satisfiability in $\mathcal{ALC}$ is NP-hard.

Hint: show the claim by reduction from SAT.

Solution: We provide a (straightforward) reduction $\varphi$ from SAT to concept satisfiability in $\mathcal{ALC}$. Given a propositional formula $f$, we obtain the $\mathcal{ALC}$ concept $\varphi(f)$ by simply viewing every propositional variable in $f$ as an atomic concept, and replacing in $f$ every occurrence of '$\forall$' with '$\wedge$', and every occurrence of '$\exists$' with '$\vee$'. Notice that $\varphi(f)$ is an $\mathcal{ALC}$ concept not containing roles.

We now show that $\varphi(f)$ is satisfiable if and only if $f$ is so.

For the “if” direction, let $f$ be satisfiable, and $\tau$ a truth value assignment such that $f\tau$ evaluates to true. We construct an interpretation $(\Delta^{\varphi}, \tau^\varphi)$ of $\varphi(f)$ as follows: $\Delta^{\varphi} = \{\sigma\}$, and for an atomic concept $A$, we set...
\[ A^{T_f} = \{ o \} \text{ if } A_T = \text{true}, \text{ and } A^{T_f} = \{ \} \text{ if } A_T = \text{false}. \] It is easy to show, by induction on the structure of \( f \), that \( \varphi(f)^{T_f} = \{ o \} \), hence \( \varphi(f) \) is satisfiable.

For the “only-if” direction, let \( \varphi(f) \) be satisfiable, \( I \) an interpretation such that \( (\varphi(f))^I \neq \emptyset \), and \( o \in (\varphi(f))^I \). We construct a truth value assignment \( \tau^f \) for \( f \) as follows: for a propositional variable \( A \) in \( f \), we set \( A_{\tau} = \text{true} \) if \( o \in A^I \), and \( A_{\tau} = \text{false} \) if \( o \notin A^I \). It is easy to show, by induction on the structure of \( f \), that \( f_{\tau^f} = \text{true} \), hence \( f \) is satisfiable. This concludes the proof.

**Exercise 6.4** Let \( q_n \), for \( n \geq 1 \), be a Boolean conjunctive query with \( n + 1 \) existential variables of the form \( \exists x_0, \ldots, x_n. P(x_0, x_1) \land P(x_1, x_2) \land \cdots \land P(x_n-1, x_n) \). Given \( n \geq 1 \):

1. construct an ALC KB \( K_n \) such that \( K_n \models q_n \).
2. construct an ALC KB \( K'_{2n} \) of size polynomial in \( n \) such that \( K'_{2n} \models q_{2n} \) and \( K'_{2n} \not\models q_{2n+1} \).

Hint: \( K'_{2n} \) “implements” a binary counter by means of \( n \) atomic concepts representing the bits of the counter, and such that the models of \( K'_{2n} \) contain a \( P \)-chain of objects of length \( 2^n \).

**Solution:**

1. There are many possible ways to construct \( K_n = \langle T_n, A_n \rangle \). We provide a few alternatives:
   - (a) \( T_n = \emptyset \) and \( A_n = \{ P(a, a) \} \);
   - (b) \( T_n = \{ A \subseteq \exists P.A \} \text{ and } A_n = \{ A(c) \} \);
   - (c) \( T_n = \emptyset \) and \( A_n = \{ P(c_0, c_1), P(c_1, c_2), \ldots, P(c_{n-1}, c_n) \} \);
   - (d) \( T_n = \{ A \subseteq \exists P.\exists P. \cdots \exists P. \exists P \} \text{ and } A_n = \{ A(c) \} \), where the number of (nested) existential restrictions in the right-hand side of the concept inclusion in \( T_n \) is equal to \( n \).
   - (e) \( T_n = \{ A \subseteq \exists P.A_1, A_1 \subseteq \exists P.A_2, \ldots, A_{n-2} \subseteq \exists P.A_{n-1}, A_{n-1} \subseteq \exists P \} \text{ and } A_n = \{ A(c) \} \).

Notice that in alternatives (a) and (b), \( T_n \) and \( A_n \) do not depend on \( n \), and work for every possible value \( n \geq 1 \).

2. We introduce \( 2n \) concepts \( B_i, \overline{B}_i, 1 \leq i \leq n \). Intuitively, \( B_i(a) \) (resp. \( \overline{B}(a) \)) says that the \( i \)-th bit of the number \( a \) is 1 (resp. 0). \( K'_{2n} = \langle T_n, A_n \rangle \), where \( T_n \) consists of the following axioms:

\[
\begin{align*}
B_i & \subseteq \exists P.\top, & 1 \leq i \leq n \\
\overline{B}_i & \subseteq \forall P.B_i \\
B_1 \sqcap \cdots \sqcap B_i \sqcap \overline{B}_{i+1} & \subseteq \forall P_i(B_1 \sqcap \cdots \sqcap B_i \sqcap B_{i+1}) & 1 \leq i \leq n - 1 \\
\overline{B}_i \sqcap \overline{B}_j & \subseteq \forall P.B_j & 1 \leq i < j \leq n \\
\overline{B}_i \sqcap B_j & \subseteq \forall P.B_j & 1 \leq i < j \leq n 
\end{align*}
\]

and \( A_n = \{ B_1(a), \ldots, B_n(a) \} \)