Knowledge Representation and Reasoning
Exercise Sheet 3

Problem 1. *A self restriction is a concept of the form* $\exists r.\mathsf{Self}$ *where* r *is a role; the semantics of self restrictions is as follows:*

$$(\exists r.\mathsf{Self})^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid (d, d) \in r^{\mathcal{I}}\}$$

Show that the extension of $\mathcal{ALC}$ *with self restrictions is more expressive than* $\mathcal{ALC}$*.*

Solution:
We only need to show that no $\mathcal{ALC}$ concept D can be equivalent to the concept $\exists r.\mathsf{Self}$. This is an easy consequence of the tree model property of $\mathcal{ALC}$. By dint of this property, D has a tree model; however, any model of $\exists r.\mathsf{Self}$ contains a loop, and thus is not a tree.

Problem 2. *Recall the following theorem given in the lectures:*

Theorem (bounded model property)
Let $\mathcal{T}$ *be an* $\mathcal{ALC}$ *TBox,* C *an* $\mathcal{ALC}$ *concept, and* $n = \mathsf{size}(\mathcal{T}) + \mathsf{size}(C)$*. If* C *has a model w.r.t.* $\mathcal{T}$*, then it has one of cardinality at most* 2^n*.*

Prove this theorem (a sketch of the proof was already given in the lecture); additionally, show that the exponential bound cannot be improved on. For this purpose, define a sequence $(\mathcal{T}_n, C_n)_{n\geq 1}$ *of* $\mathcal{ALC}$ *TBoxes* $\mathcal{T}_n$ *and concepts* C_n *such that*

1. *the sizes of* $\mathcal{T}_n$ *and* C_n *are polynomial in* n*; and*
2. *no model of* C_n *with respect to* $\mathcal{T}_n$ *can contain less than* 2^n *elements.*

Solution:
Proof of theorem:
Let $\mathcal{I}$ be a model of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$, and $S = \mathsf{sub}(\mathcal{T}) \cup \mathsf{sub}(C)$. Then we have $|S| \leq n$, and thus the domain of the S-filtration $\mathcal{J}$ of $\mathcal{I}$ satisfies $|\Delta^{\mathcal{J}}| \leq 2^n$ by Lemma 6.9 (see lecture notes). Thus, it remains to show that $\mathcal{J}$ is a model of C w.r.t. $\mathcal{T}$.

Let $d \in \Delta^{\mathcal{I}}$ be such that $d \in C^{\mathcal{I}}$. Since $C \in S$, we know that $d \in C^{\mathcal{I}}$ implies $[d]_S \in C^{\mathcal{J}}$ by Lemma 6.11 (see lecture notes), and thus $C^{\mathcal{J}} \neq \emptyset$. In addition, it is easy to see that $\mathcal{J}$ is a model of $\mathcal{T}$. In fact, let $D \sqsubseteq E$ be a GCI in $\mathcal{T}$, and $[e]_S \in D^{\mathcal{J}}$. We must show $[e]_S \in E^{\mathcal{J}}$. Since $D \in S$, Lemma 6.11 yields $e \in D^{\mathcal{I}}$, and thus $e \in E^{\mathcal{I}}$ since $\mathcal{I}$ is a model of $\mathcal{T}$. But then $E \in S$ implies $[e]_S \in E^{\mathcal{J}}$, again by Lemma 6.11.

Tightness of bound:
Even without a TBox we can construct concept descriptions C_n ($n \geq 1$) such that, for each n, the models of C_n "contain" a full binary tree of depth n whose leaves are labeled with the 2^n disjoint concepts $\neg A_0 \sqcap \ldots \sqcap \neg A_{n-1}$, $A_0 \sqcap \neg A_1 \sqcap \ldots \sqcap \neg A_{n-1}$, $\neg A_0 \sqcap A_1 \sqcap \neg A_2 \ldots \sqcap \neg A_{n-1}$, $\ldots$, $A_0 \sqcap \ldots \sqcap A_{n-1}$, which can be seen as binary representations of the numbers $0, 1, 2, \ldots, 2^n - 1$:

$$\begin{aligned} C_n \quad = \quad & \textstyle\bigsqcap_{0 \leq i \leq n-1} (\forall r.)^i (\exists r.A_i \sqcap \exists r.\neg A_i) \sqcap \\ & \textstyle\bigsqcap_{0 \leq i < j \leq n-2} (\forall r.)^j ((A_i \to \forall r.A_i) \sqcap (\neg A_i \to \forall r.\neg A_i)). \end{aligned}$$

Basically, the first conjunct of C_n (first line) ensures that an individual d at distance i from an element of C_n (i.e., one that can be reached from an element of C_n by an r-chain of length i) has two distinct r-successors d_1 and d_2, one belonging to A_i and one not belonging to A_i. In addition, the "decision" taken at distance i (A_i or $\neg A_i$) stays the same for all r-successors of d_1 and d_2, up to distance $n-1$. For this reason, any individual belonging to C_n has at distance $n-1$ at least 2^n r-successors labeled with binary encodings of the numbers $0, 1, 2, \ldots, 2^n - 1$. Note that this construction is similar to the one used in Chapter 5 to show PSpace-hardness of satisfiability in $\mathcal{ALC}$ without TBoxes.

With a TBox, we can even enforce a sequence of 2^n elements, consecutively labeled with concepts encoding the numbers $0, 1, 2, \ldots, 2^n - 1$. Let $C_n = \neg A_0 \sqcap \ldots \sqcap \neg A_{n-1}$ and the TBox $\mathcal{T}_n$ consist of the following GCIs: for all $0 \leq i < n-1$

$$\begin{aligned} \neg A_i \sqcap \textstyle\bigsqcap_{0 \leq j < i} A_i \to \; & \exists r.\top \sqcap \\ & \forall r.(A_i \sqcap \textstyle\bigsqcap_{0 \leq j < i} \neg A_i) \sqcap \\ & \textstyle\bigsqcap_{i+1 \leq j < n} ((A_j \to \forall r.A_j) \sqcap (\neg A_j \to \forall r.\neg A_j)). \end{aligned}$$

Basically, the TBox realizes a binary counter. In a model of C_n w.r.t. $\mathcal{T}_n$, we start with an element of C_n, which belongs to the concept encoding the number 0. As long as the number $2^n - 1$ is not yet reaches (i.e., one of the digits is still 0, corresponding to one of the A_i's not being satisfied by the current individual), an r-successor is generated that belongs to the next number. This enforces the existence of a role change of length 2^n, on which the individuals belong the the encodings of the numbers $0, 1, 2, \ldots, 2^n - 1$, and are such all different.

Problem 3. *Consider the following $\mathcal{ALC}$-concept C:*

$$C = \exists S.A \sqcap \forall S.(\neg A \sqcup \neg B) \sqcap \exists R.A \sqcap \forall R.B$$

Using the tableau algorithm for concept satisfiability show that C is satisfiable. Write down a finite tree interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$.

Solution:
Proceed as follows.

- The algorithm is initialised with $\mathcal{A}_0 = \{a : C\}$.
- We can apply the $\sqcap$-rule three times to get $\mathcal{A}_3 = \mathcal{A} \cup \{a : \exists S.A, a : \forall S.(\neg A \sqcup \neg B), a : \exists R.A, a : \forall R.B\}$
- We can then apply the $\exists$-rule to $a : \exists S.A \in \mathcal{A}_3$ to get $\mathcal{A}_4 = \mathcal{A}_3 \cup \{(a,b) : S, b : A\}$
- Then, we can apply the $\forall$-rule to $\{a : \forall S.(\neg A \sqcup \neg B), (a,b) : S\} \subseteq \mathcal{A}_4$ to get $\mathcal{A}_5 = \mathcal{A}_4 \cup \{b : (\neg A \sqcup \neg B)\}$
- Next, we apply the $\sqcup$ rule to $b : (\neg A \sqcup \neg B) \in \mathcal{A}_5$. If we choose $\mathcal{A}_6 = \mathcal{A}_5 \cup \{b : \neg A\}$, then we immediately get a clash and have to backtrack and choose $\mathcal{A}_6 = \mathcal{A}_5 \cup \{b : \neg B\}$.
- Next, apply the $\exists$-rule to $a : \exists R.A \in \mathcal{A}_6$ to get $\mathcal{A}_7 = \mathcal{A}_6 \cup \{(a,c) : R, c : A\}$.
- Finally, we apply the $\forall$-rule to $\{a : \forall R.B, (a,c) : R\} \subseteq \mathcal{A}_7$ to get $\mathcal{A}_8 = \mathcal{A}_7 \cup \{c : B\}$, with $\mathcal{A}_8$ being a complete and clash-free ABox.

The interpretation follows trivially from the ABox, and it is clear that $a^{\mathcal{I}} \in C^{\mathcal{I}}$, so $C^{\mathcal{I}} \neq \emptyset$. Finally, it is important to note that the rules could have been applied in a different order; the fact that the algorithm returns `true` given C and that we can obtain the required tree model from the final clash-free completion graph is independent from the order of rule applications.

Problem 4. *Consider again the concept C from Problem 3.*

1. *Using the tableau algorithm for concept satisfiability w.r.t. a TBox show that C is satisfiable w.r.t. the TBox $\mathcal{T} = \{A \sqsubseteq \exists R.A\}$. To reduce non-determinism use the lazy unfolding version of the $\sqsubseteq$-rule rather than the general KB version of the rule — this isn't correct in general as $A \sqsubseteq \exists R.A$ is cyclical, but it will work in this case.*
2. *Write down a finite (possibly non-tree) model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$.*
3. *Specify a (possibly infinite) tree model $\mathcal{I}'$ of $\mathcal{T}$ such that $C^{\mathcal{I}'} \neq \emptyset$.*

Solution:
The algorithm can proceed exactly as in Problem 3 to construct $\mathcal{A}_8$. We can then proceed as follows:

- apply the $\sqsubseteq$-rule twice to b and c to get $\mathcal{A}_{10} = \mathcal{A}_8 \cup \{b : \exists R.A, c : \exists R.A\}$.
- Apply the $\exists$-Rule twice to get $\mathcal{A}_{12} = \mathcal{A}_{10} \cup \{(b,d) : R, d : A, (c,e) : R, e : A\}$.
- apply the $\sqsubseteq$-rule twice more to d and e to get $\mathcal{A}_{14} = \mathcal{A}_{12} \cup \{d : \exists R.A, e : \exists R.A\}$.

Now, d is blocked by b and e is blocked by c so $\mathcal{A}_{14}$ is complete and clash-free. Note that, again, the order of rule applications is irrelevant. From $\mathcal{A}_{14}$ we can construct an ABox $\mathcal{A}'$ by replacing $(b,d):R$ and $(c,e):R$ with "loop-back" role assertions $(b,b):R$ and $(c,c):R$, and by removing $d:\exists R.A$ and $e:\exists R.A$. The finite model follows trivially from this ABox. To obtain a tree model we can "unravel" the loops to create two infinite R-chains connecting copies of b and c respectively.

Problem 5. *The description logic $\mathcal{ALCH}$ is obtained from $\mathcal{ALC}$ by also allowing* role inclusion axioms *in the TBox. A role inclusion axiom is of the form $R \sqsubseteq S$ where R and S are atomic roles. An interpretation $\mathcal{I}$ satisfies a role inclusion axiom $R \sqsubseteq S$ if $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$.*

1. *Provide a semantics for $\mathcal{ALCH}$ via translation to FOL by extending the one for $\mathcal{ALC}$ given in the Lecture Notes.*
2. *Modify the tableau algorithm for $\mathcal{ALC}$ concept satisfiability w.r.t. a TBox given in the Lecture Notes to support $\mathcal{ALCH}$.*
3. *Use such modified algorithm to show that the concept C from Problem 3 is unsatisfiable w.r.t. the TBox $\mathcal{T} = \{A \sqsubseteq \exists R.A, S \sqsubseteq R\}$.*

Solution:
The semantics via translation to FOL of role inclusion axioms is straightforward. $\mathcal{ALCH}$-concepts coincide with $\mathcal{ALC}$-concepts. So, we only need to define the transformation for the new type of TBox axiom we are introducing:

$$\pi(R \sqsubseteq S) = \forall x.\forall y.(R(x,y) \to S(x,y))$$

In order to support $\mathcal{ALCH}$ the easiest modification to the tableau algorithm from the Lecture Notes is to add a new completion rule as follows:

$\sqsubseteq_R$-rule: if 1. $(a,b):R \in \mathcal{A}$, $R \sqsubseteq S \in \mathcal{T}$, and
2. $(a,b):S \notin \mathcal{A}$
then $\mathcal{A} \longrightarrow \mathcal{A} \cup \{(a,b):S\}$

The algorithm can proceed exactly as in Problem 4 to construct $\mathcal{A}_{14}$, but the construction can continue as follows:

- apply the $\sqsubseteq_R$-rule to $(a,b):S \in \mathcal{A}_{14}$ to give $\mathcal{A}_{15} = \mathcal{A}_{14} \cup \{(a,b):R\}$.
- apply the $\forall$-rule to $\{a:\forall R.B, (a,b):R\} \subseteq \mathcal{A}_{15}$ to get $\mathcal{A}_{16} = \mathcal{A}_{15} \cup \{c:B\}$.

We now have a clash, and no further backtracking possibilities, so the algorithm will return "inconsistent".

Problem 6. *Consider the First Order Logic (FOL) knowledge base $\mathcal{K}$ consisting of the following sentences:*

$$\forall x.(\mathsf{Man}(x) \leftrightarrow \mathsf{Human}(x) \land \mathsf{Male}(x)) \tag{1}$$
$$\forall x.(\mathsf{Parent}(x) \leftrightarrow \mathsf{Human}(x) \land \exists y.(\mathsf{hasChild}(x, y) \land \mathsf{Human}(y))) \tag{2}$$
$$\forall x.(\mathsf{Father}(x) \leftrightarrow \mathsf{Man}(x) \land \exists y.(\mathsf{hasChild}(x, y) \land \mathsf{Human}(y))) \tag{3}$$
$$\forall x.(\mathsf{GrandFather(x)} \leftrightarrow \mathsf{Man}(x) \land \exists y.(\mathsf{hasChild}(x, y) \land \mathsf{Parent}(y))) \tag{4}$$

Do the following:

- *Write an $\mathcal{ALC}$ TBox $\mathcal{T}$ that is logically equivalent to $\mathcal{K}$.*
- *Determine whether the axiom*

$$\mathsf{GrandFather} \sqsubseteq \mathsf{Parent}$$

  is a logical consequence of $\mathcal{T}$ by applying the $\mathcal{ALC}$ tableau algorithm for concept satisfiability w.r.t. a TBox. (You can apply optimisations to reduce the number of rule applications).

Solution:
The following TBox $\mathcal{T}$ is equivalent to $\mathcal{K}$:

$$\begin{aligned}
\mathsf{Man} &\equiv \mathsf{Human} \sqcap \mathsf{Male}\\
\mathsf{Parent} &\equiv \mathsf{Human} \sqcap \exists \mathsf{hasChild}.\mathsf{Human}\\
\mathsf{Father} &\equiv \mathsf{Man} \sqcap \exists \mathsf{hasChild}.\mathsf{Human}\\
\mathsf{GrandFather} &\equiv \mathsf{Man} \sqcap \exists \mathsf{hasChild}.\mathsf{Parent}
\end{aligned}$$

This TBox is *unfoldable* according to the definition of unfoldability given in the slides. Hence, we do not need to apply the TBox rule and it suffices to use lazy unfolding.

To show that $\mathcal{T} \models \mathsf{GrandFather} \sqsubseteq \mathsf{Parent}$ using a tableau algorithm, we need to check whether the concept $C = \mathsf{GrandFather} \sqcap \neg\mathsf{Parent}$ is (un)satisfiable w.r.t. $\mathcal{T}$. We therefore initialise $\mathcal{A}_0 = \{a : C\}$ and proceed with tableau expansion as per the above examples. It is easy to see that all expansion choices lead to a clash, and that the entailment thus holds.