COMP321 (Ontology Languages): Test 1

Lecturer: F. Wolter
Time: 50 minutes

This test makes up 10 percent of the final mark for this module. You can achieve 100 marks.

1. Consider the following $\mathcal{EL}$-TBox $\mathcal{T}$:

```
SmallProject $\sqsubseteq$ Project
\exists manages.Project $\sqsubseteq$ Manager
JuniorManager $\sqsubseteq$ \exists manages.SmallProject
JuniorManager $\sqcap$ Manager $\sqsubseteq$ Employee
```

- Apply the $\mathcal{EL}$-subsumption algorithm from the Comp321-Slides to compute $S(A)$ for every concept name $A$ in $\mathcal{T}$ and $R(r)$ for every role name $r$ of $\mathcal{T}$. In your answer, show how the rules (simpleR), (conjR), (rightR) and (leftR) are applied step-by-step in the computation of $S$ and $R$.

- Using $S$, determine whether Manager $\sqsubseteq_\mathcal{T}$ Employee (i.e., whether Manager is subsumed by Employee w.r.t. $\mathcal{T}$).

- Using $S$, determine whether JuniorManager $\sqsubseteq_\mathcal{T}$ Manager (i.e., whether JuniorManager is subsumed by Manager w.r.t. $\mathcal{T}$).

- Using $S$, determine whether JuniorManager $\sqsubseteq_\mathcal{T}$ Employee (i.e., whether JuniorManager is subsumed by Employee w.r.t. $\mathcal{T}$). (30 marks)

2. Consider the interpretation $\mathcal{I}$ defined by

- $\Delta^\mathcal{I} = \{a, b, c\}$;
- $A^\mathcal{I} = \{a, b\}$, $B^\mathcal{I} = \{c\}$, $r^\mathcal{I} = \{(a, b), (a, c), (b, b)\}$.

Determine the following sets:

- $(\exists r.A)^\mathcal{I}$;
- $(\forall r.A)^\mathcal{I}$;
- $(\exists r.B)^\mathcal{I}$;
- $(\forall r.B)^\mathcal{I}$. (20 marks)
3. Consider the \( \mathcal{ALC} \)-TBox \( \mathcal{T} \):

\[
\text{Manager} \subseteq \exists \text{works_for.Employer} \\
\text{Manager} \subseteq \exists \text{manages.Unit} \\
\text{Manager} \subseteq \forall \text{manages.Department} \\
\text{Employer} \subseteq \exists \text{registered_at.TaxOffice}
\]

- Translate \( \mathcal{T} \) into natural language (English).
- Define a model \( \mathcal{I} \) of \( \mathcal{T} \) in which Manager is satisfied. (In other words, define an interpretation \( \mathcal{I} \) such that \( \mathcal{I} \models \mathcal{T} \) and \( \text{Manager}^\mathcal{I} \neq \emptyset \).)
- Add \( \text{Unit} \sqcap \text{Department} \subseteq \bot \) to \( \mathcal{T} \). Is the resulting TBox satisfiable? Provide an (informal!) argument for your answer. \( \text{(20 marks)} \)

4. Consider the \( \mathcal{ALC} \)-concept

\[
C = (\neg A) \sqcap (\exists r.\exists r.A) \sqcap (\forall r.\neg A)
\]

Apply the \( \mathcal{ALC} \)-tableau algorithm from the Comp321-Slides to the concept \( C \) to determine whether \( C \) is satisfiable or not. In your answer, show how the rules \( \rightarrow \sqcap, \rightarrow \sqcup, \rightarrow \exists \), and \( \rightarrow \forall \) are applied step-by-step to the constraint system \( x : C \). If \( C \) is satisfiable, construct an interpretation \( \mathcal{I} \) satisfying \( C \) (i.e., with \( C^\mathcal{I} \neq \emptyset \)). \( \text{(30 marks)} \)

**Solution for 1.**

The initial assignment is given by

\[
\begin{align*}
S(\text{SmallProject}) &= \{\text{SmallProject}\} \\
S(\text{Project}) &= \{\text{Project}\} \\
S(\text{Manager}) &= \{\text{Manager}\} \\
S(\text{Employee}) &= \{\text{Employee}\} \\
S(\text{JuniorManager}) &= \{\text{JuniorManager}\} \\
R(\text{manages}) &= \emptyset
\end{align*}
\]

Now applications of (simpleR), (conjR), (rightR), (leftR) are step-by-step as follows:

- Update \( S \) using (simpleR):
  \[
  S(\text{SmallProject}) = \{\text{Project, Smallproject}\}
  \]

- Update \( R \) using (rightR) for the third inclusion of \( \mathcal{T} \):
  \[
  R(\text{manages}) = \{(\text{JuniorManager, SmallProject})\}.
  \]
• Update $S$ using (leftR) for the second inclusion of $T$:

$$S(\text{JuniorManager}) = \{\text{JuniorManager, Manager}\}$$

• Update $S$ using (conjR) for the last inclusion:

$$S(\text{JuniorManager}) = \{\text{JuniorManager, Manager, Employee}\}.$$  

The final assignment is

$$S(\text{SmallProject}) = \{\text{SmallProject, Project}\}$$
$$S(\text{Project}) = \{\text{Project}\}$$
$$S(\text{Manager}) = \{\text{Manager}\}$$
$$S(\text{Employee}) = \{\text{Employee}\}$$
$$S(\text{JuniorManager}) = \{\text{JuniorManager, Manager, Employee}\}$$
$$R(\text{manages}) = \{(\text{JuniorManager, SmallProject})\}.$$  

• Manager $\sqsubseteq_T \text{Employee}$ does not hold since $\text{Employee} \notin S(\text{Manager})$.

• JuniorManager $\sqsubseteq_T \text{Manager}$ holds since $\text{Manager} \in S(\text{JuniorManager})$.

• JuniorManager $\sqsubseteq_T \text{Employee}$ holds since $\text{Employee} \in S(\text{JuniorManager})$.

**Solution for 2.**

Consider the interpretation $I$ defined by

• $\Delta^I = \{a, b, c\};$

• $A^I = \{a, b\};$

• $B^I = \{c\};$

• $r^I = \{(a, b), (a, c), (b, b)\}.$

We have:

• $(\exists r.A)^I = \{a, b\};$

• $(\forall r.A)^I = \{b, c\};$

• $(\exists r.B)^I = \{a\};$

• $(\forall r.B)^I = \{c\}.$

**Solution for 3.**

• Every manager works for an employer;

• Every manager manages a unit.
• Every manager only manages departments;
• Every employer is registered at some tax office.

An interpretation \( I \) that is a model of \( T \) with \( \text{Manager}^I \neq \emptyset \) is given by setting \( \Delta^I = \{a, b, c, d\} \) and

- \( \text{Manager}^I = \{a\} \);
- \( \text{Employer}^I = \{b\} \);
- \( \text{Unit}^I = \{c\} \);
- \( \text{Department}^I = \{c\} \);
- \( \text{TaxOffice}^I = \{d\} \);
- \( \text{registered} \_ \text{at}^I = \{(b, d)\} \);
- \( \text{manages}^I = \{(a, c)\} \).

The resulting TBox is satisfiable. For example, every \( I \) in which \( \text{Manager}^I = \emptyset \) and \( \text{Employer}^I = \emptyset \) is a model of the extended TBox. Note, however, that \( \text{Manager} \) is not satisfiable anymore because \( \text{Unit} \) and \( \text{Department} \) are disjoint (according to the new inclusion) but every manager manages a unit and any such unit is, by the value restriction in the third inclusion, a department.

Solution for 4.

Let

\[ C = (\neg A) \cap (\exists r. \exists r. A) \cap (\forall r. \neg A) \]

As \( C \) is already in negation normal form, the tableaux algorithm starts with

\[ S_0 = \{x : (\neg A) \cap (\exists r. \exists r. A) \cap (\forall r. \neg A)\} \]

An application of the rule \( \rightarrow \exists \) gives

\[ S_1 = S_0 \cup \{x : \neg A, x : (\exists r. \exists r. A) \cap (\forall r. \neg A)\} \]

An application of the rule \( \rightarrow \forall \) gives

\[ S_2 = S_1 \cup \{x : (\exists r. \exists r. A), x : (\forall r. \neg A)\} \]

An application of the rule \( \rightarrow \exists \) gives

\[ S_3 = S_2 \cup \{(x, y) : r, y : \exists r. A\} \]

An application of the rule \( \rightarrow \exists \) gives

\[ S_4 = S_3 \cup \{(y, z) : r, z : A\} \]

An application of the rule \( \rightarrow \forall \) gives

\[ S_5 = S_4 \cup \{y : \neg A\} \]

No rule is applicable to \( S_5 \) and \( S_5 \) does not contain any clash. Thus, \( C \) is satisfiable.

An interpretation \( I \) satisfying \( C \) is given by
• $\Delta^I = \{x, y, z\}$;
• $A^I = \{z\}$;
• $r^I = \{(x, y), (y, z)\}$.  