Assignment 3

Course Lecturer: Yizheng Zhao

April 25, 2023

⋆ This assignment, due on May 11th, contributes to 10% of the final mark for this course. In addition, you have the opportunity to earn up to three bonus marks if you successfully complete the last three questions. These bonus marks can potentially increase a student's overall marks, but they are subject to a maximum total mark of 100 for the course. By offering these bonus marks, we are providing an incentive for students to strive for excellence and rewarding those who demonstrate exceptional understanding of the course material.

Question 1. Termination

Let E be an $\mathcal{ALC}$-concept. By $\#E$ we denote the number of occurrences of the constructors $\sqcap$, $\sqcup$, $\exists$, $\forall$ in E. The multiset $M(E)$ contains, for each occurrence of a subconcept of the form $\neg F$ in E, the number $\#F$.

- Following this representation, prove that exhaustively applying the transformations below to an $\mathcal{ALC}$ concept always terminates, regardless of the order of rule application:

$$\begin{aligned}
\neg(E \sqcap F) &\rightsquigarrow \neg\neg\neg E \sqcup \neg\neg\neg F \\
\neg(E \sqcup F) &\rightsquigarrow \neg\neg\neg E \sqcap \neg\neg\neg F \\
\neg\neg E &\rightsquigarrow E \\
\neg(\exists r.E) &\rightsquigarrow \forall r.\neg E \\
\neg(\forall r.E) &\rightsquigarrow \exists r.\neg E
\end{aligned}$$

Question 2. Negation Normal Norm (NNF)

Let $\mathcal{T}$ be an acyclic TBox in NNF. $\mathcal{T}^{\sqsubseteq}$ is obtained from $\mathcal{T}$ by replacing each concept definition $A \equiv C$ with the concept inclusion $A \sqsubseteq C$.

- Prove that every concept name is satisfiable w.r.t. $\mathcal{T}$ iff it is satisfiable w.r.t. $\mathcal{T}^{\sqsubseteq}$. Does this also hold for the acyclic TBox $\{A \equiv C \sqcap \neg B, B \equiv P, C \equiv P\}$?

Question 3. Tableau Algorithm for ABoxes with Acyclic TBoxes

We consider the tableau algorithm consistent$(\mathcal{T}, \mathcal{A})$ for acyclic TBoxes $\mathcal{T}$, which is obtained from consistent$(\mathcal{A})$ by adding the $\equiv_1$-rule and the $\equiv_2$-rule for unfolding $\mathcal{T}$.

- Prove that consistent$(\mathcal{T}, \mathcal{A})$ is a decision procedure for the consistency of $\mathcal{ALC}$-knowledge bases with acyclic TBoxes.

Question 4. Tableau Algorithm for ABoxes with Acyclic TBoxes

Use the Tableau algorithm consistent$(\mathcal{T}, \mathcal{A})$ for acyclic TBoxes to determine whether the subsumption

$$\neg(\forall r.A) \sqcap \forall r.C \sqsubseteq_{\mathcal{T}} \forall r.E$$

holds w.r.t. the acyclic TBox

$$\mathcal{T} = \{C \equiv (\exists r.\neg B) \sqcap \neg A, D \equiv \exists r.B, E \equiv \neg(\exists r.A) \sqcap \exists r.D\}.$$

Question 5. Anywhere Blocking

We consider a different form of blocking, which allows individuals to be blocked by individuals who are not necessarily their ancestors, known as anywhere blocking. This approach employs an individual a's age, denoted as $\mathsf{age}(a)$, to determine the blocking relationship, instead of relying on the ancestor relation.

The age of an individual is defined as 0 for individuals that occur in the input ABox $\mathcal{A}$, while a new individual generated by the nth application of the $\exists$-rule is assigned an age of n. This approach expands the scope of blocking beyond the ancestor relation, enabling individuals to be blocked based on their age, which could result in more effective blocking in certain situations.

Let $\mathcal{A}'$ be an ABox obtained by applying the tableau rules of consistent$(\mathcal{T}, \mathcal{A})$ for general TBoxes. A tree individual b is anywhere blocked by an individual a in $\mathcal{A}'$ if

- $\mathsf{con}_{\mathcal{A}'}(b) \subseteq \mathsf{con}_{\mathcal{A}'}(a)$,
- $\mathsf{age}(a) < \mathsf{age}(b)$, and
- a is not blocked.

As before, the descendants of b are then also considered blocked.

- Prove that the tableau algorithm with anywhere blocking is a decision procedure for the consistency of $\mathcal{ALC}$-knowledge bases with general TBoxes.

Question 6. Precompletion of Tableau Algorithm

We consider an $\mathcal{ALC}$-knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with $\mathcal{T}$ being a general TBox. A *precompletion* of $\mathcal{K}$ is a clash-free ABox $\mathcal{A}$ obtained from $\mathcal{K}$ by exhaustively applying all expansion rules except the $\exists$-rule.

- Prove that $\mathcal{K}$ is consistent if, and only if, there is a precompletion $\mathcal{A}$ of $\mathcal{K}$ such that, for all individual names a occurring in $\mathcal{A}$, the concept description $C^a_{\mathcal{A}} := \bigsqcap_{a:C \in \mathcal{A}} C$ is satisfiable w.r.t. $\mathcal{T}$.

Question 7. Tableau Algorithm for $\mathcal{ALCN}$

- Prove soundness and completeness of the Tableau algorithm for $\mathcal{ALCN}$ presented in the lecture slides.

Question 8. Tableau Algorithm for $\mathcal{ALCQ}$

We extend the Tableau algorithm from $\mathcal{ALCN}$ to $\mathcal{ALCQ}$ by modifying the $\geq$-rule and the $\leq$-rule as follows:

The $\geq$-rule	
Condition:	$\mathcal{A}$ contains $a:(\geq n\, r.C)$, but there are no n distinct individuals $b_1, \ldots, b_n$ with $\{(a, b_i):r,\ b_i:C \mid 1 \leq i \leq n\} \subseteq \mathcal{A}$, and a is not blocked
Action:	$\mathcal{A} \longrightarrow \mathcal{A} \cup \{(a, d_i):r,\ d_i:C \mid 1 \leq i \leq n\} \cup \{d_i \neq d_j \mid 1 \leq i < j \leq n\}$, where $d_1, \ldots, d_n$ are new individual names

The $\leq$-rule	
Condition:	$\mathcal{A}$ contains $a:(\leq n\, r.C)$, and there are $n+1$ distinct individuals $b_0, \ldots, b_n$ with $\{(a, b_i):r,\ b_i:C \mid 0 \leq i \leq n\} \subseteq \mathcal{A}$
Action:	$\mathcal{A} \longrightarrow \mathsf{prune}(\mathcal{A}, b_j)[b_j \mapsto b_i] \cup \{b_i = b_j\}$ for $i \neq j$ such that, if b_j is a root individual, then so is b_i

- For the knowledge base

$$(\{C \sqsubseteq E\}, \{a : \leq 1 r.(D \sqcap E), (a, b) : r, b : C \sqcap D, (a, c) : r, c : D \sqcap E, c : \neg C\}),$$

determine whether it is consistent, and whether the proposed algorithm detects this.

Question 9. A Complex in $\mathcal{ALC}$ Extensions
The DL $\mathcal{S}$ extends $\mathcal{ALC}$ with *transitivity axioms* $\mathsf{trans}(r)$ for role names $r \in \mathsf{R}$. Their semantics is defined as follows: $\mathcal{I} \models \mathsf{trans}(r)$ iff $r^{\mathcal{I}}$ is transitive. Furthermore, an $\mathcal{S}$ knowledge base $\mathcal{K} := (\mathcal{T}, \mathcal{A}, \mathcal{R})$ consists of an $\mathcal{ALC}$ knowledge base $(\mathcal{T}, \mathcal{A})$, and an additional RBox $\mathcal{R}$ of transitivity axioms. Prove the following:

- For an arbitrary TBox $\mathcal{T}$, the concept $C_{\mathcal{T}}$ is defined as $\bigsqcap_{C \sqsubseteq D \in \mathcal{T}} \neg C \sqcup D$. Then $\mathcal{T}$ and $\mathcal{T}' = \{\top \sqsubseteq C_{\mathcal{T}}\}$ have the same models.

- Let $\mathcal{K} := \{\mathcal{T}, \mathcal{A}, \mathcal{R}\}$ be a knowledge base such that, without loss of generality, $\mathcal{T}$ consists of a single GCI $\top \sqsubseteq C_{\mathcal{T}}$, and $C_{\mathcal{T}}$ is in NNF. Define the $\mathcal{ALC}$ knowledge base $\mathcal{K}^+ := (\mathcal{T}^+, \mathcal{A})$ where

$$\mathcal{T}^+ := \mathcal{T} \cup \{\forall r.C \sqsubseteq \forall r.\forall r.C \mid \mathsf{trans}(r) \in \mathcal{R} \text{ and } \forall r.C \in \mathsf{Sub}(C_{\mathcal{T}})\}.$$

  Then $\mathcal{K}$ is consistent, if and only if, $\mathcal{K}^+$ is consistent. Consequently, the Tableau algorithm for $\mathcal{ALC}$ can also be used for $\mathcal{S}$.

Question 10 (with 1 bonus mark). Pushdown automata
Let a k-PDA be a pushdown automaton that has k stacks. Thus a 0-PDA is an NFA and a 1-PDA is a conventional PDA. You already know that 1-PDAs are more powerful (recognize a larger class of languages) than 0-PDAs.

- (a) Show that 2-PDAs are more powerful than 1-PDAs.

- (b) Show that 3-PDAs are *not* more powerful than 2-PDAs.

Question 11 (with 1 bonus mark). Turing machine

- Define a standard TM that decides the following language:

$$L = \{m \in \{0,1\}^* \mid |m|_0 = |m|_1\}$$

  where $|m|_0$ and $|m|_1$ denote respectively the number of 0's and 1's in m.

Question 12 (with 1 bonus mark). Decidability
Let A be the language containing only the single string s, where

$$\mathsf{s} = \begin{cases} 0 & \text{if reasonably-priced yummies will never be found in the canteens of NJU} \\ 1 & \text{if reasonably-priced yummies will be found in the canteens of NJU someday} \end{cases}$$

- (a) Is A decidable? Why or why not? For the sake of this question, assume that whether reasonably-priced yummies will be found in the canteens of NJU has an unambiguous YES or NO answer.