KRP — Assignment 3

Student: Qiankun Ji

June 2, 2024

⋆ **This assignment, due on 12th May at 23:59, contributes to 10% of the final marks for this course. Please be advised that only Questions 1 — 10 are mandatory. Nevertheless, students can earn up to one bonus mark by completing Question 11. This bonus mark can potentially augment a student's overall marks but is subject to a maximum total of 100 for the course. By providing bonus marks, we aim to incentivize students to excel in their studies and reward those with a remarkable grasp of the course materials.**

Question 1. Basic Tableau Algorithm

- Apply the Tableau algorithm consistent($\mathcal{A}$) to the following ABox:

  $$\mathcal{A} = \{(b,a) : r, (a,b) : r, (a,c) : s, (c,b) : s, a : \exists s.A, b : \forall r.((\forall s.\neg A) \sqcup (\exists r.B)), c : \forall s.(B \sqcap (\forall s.\bot))\}.$$

  If $\mathcal{A}$ is consistent, draw the model generated by the algorithm.

My Solution Question 1. Apply the Tableau algorithm step by step:

1. Start with the set of concepts and roles:
- Concepts: A, B.
- Roles: r, s.

2. Introduce individuals from the ABox: a, b, c.

3. Add the concept assertions to the tableau:
- a is an instance of $\exists s.A$.
- b is an instance of $\forall r.((\forall s.\neg A) \sqcup (\exists r.B))$.
- c is an instance of $\forall s.(B \sqcap (\forall s.\bot))$.

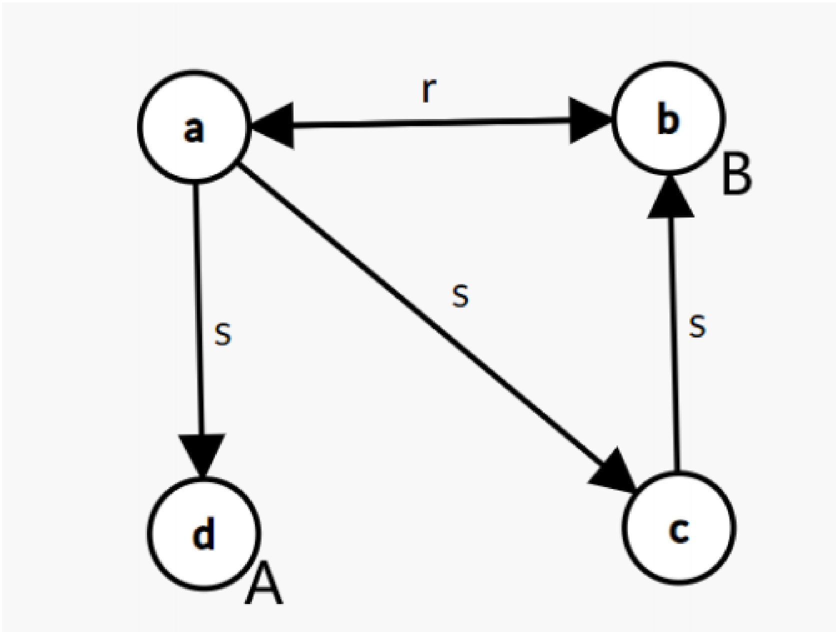
4. Add the role assertions to the tableau: - (b,a) is related via r.
- (a,b) is related via r.
- (a,c) is related via s.
- (c,b) is related via s.

5. Apply the expansion rules until no new information can be added or a contradiction arises:
- For $a : \exists s.A$, add $A(a)$ to the tableau.
- For $b : \forall r.((\forall s.\neg A) \sqcup (\exists r.B))$, add $\neg A(b)$ and $B(b)$ to the tableau for every r-successor of b. Since (b,a) and (a,b) are related via r, add $\neg A(a)$ and $B(a)$ to the tableau for b.
- For $c : \forall s.(B \sqcap (\forall s.\bot))$, add $B(c)$ and $\bot$ to the tableau for every s-successor of c. Since (c,b) is related via s, add $B(b)$ and $\bot$ to the tableau for c.

Above all, we can conclude that $\mathcal{A}$ **is consistent**.And a model $\mathcal{I}$ of $\mathcal{A}$ that fits:

$$\Delta^{\mathcal{I}} = \{a, b, c, d\}$$
$$A^{\mathcal{I}} = \{d\}, B^{\mathcal{I}} = \{b\}$$
$$a^{\mathcal{I}} = a, b^{\mathcal{I}} = b, c^{\mathcal{I}} = c, d^{\mathcal{I}} = d$$
$$r^{\mathcal{I}} = \{(b, a) : r, (a, b) : r, (a, c) : s, (c, b) : s, (a, d) : s\}$$

draw the model:



Question 2. Modification of Tableau Algorithm

We consider an $\mathcal{ALC}$ TBox $\mathcal{T}$ consisting only of the following two kinds of axioms:

- role inclusions of the form $r \sqsubseteq s$, and
- role disjointness constraints of the form $\mathsf{disjoint}(r, s)$.

where r and s are role names. An interpretation $\mathcal{I}$ satisfies these axioms if

- $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$, and
- $r^{\mathcal{I}} \cap s^{\mathcal{I}} = \emptyset$, respectively.

Modify the Tableau algorithm $\mathsf{consistent}(\mathcal{A})$ to decide the consistency of $(\mathcal{T}, \mathcal{A})$, where $\mathcal{A}$ is an ABox and $\mathcal{T}$ a TBox containing only role inclusions and role disjointness constraints. Show that the algorithm remains terminating, sound, and complete.

My Solution Question 2. Add this rule: $\sqsubseteq -rule$

Condition: $\mathcal{A}$ contains $(a, b) : r$, but not $(a, b) : s$. At the same time, $\mathcal{T}$ contains $r \subseteq s$.

Action: $\mathcal{A} \to \mathcal{A} \cup \{(a, b) : s\}$.

Modify clash-definition: $(\mathcal{T}, \mathcal{A})$ contains a clash iff $\{a : C, a : \neg C\} \subseteq \mathcal{A}$ or $\{(a, b) : r, (a, b) : s\} \subseteq \mathcal{A}$ and $\{disjoint(r, s)\} \subseteq \mathcal{T}$

Termination:

(1). Because my change don't add new concept assertion compared to the original Tableau algorithm, so $|con\mathcal{A}(a)| \leq m$ ($m = |sub(\mathcal{A})|$) also holds for any individual name a.

(2). Because my change add new role assertion without new individual name. And we know every role assertion in $\mathcal{A}$ can generate at most $|\mathcal{T}|$ new role assertion with the "$\sqsubseteq -rule$". So, there exists at most$|\mathcal{T}|m$ role assertion after all "expand" processes.

(3). Because my change don't add new individual name compared to the original Tableau algorithm. So the words "a given individual name can cause the addition of at most m new individual names" holds too, which means the outdegree of each tree in the forest-shaped ABox is thus bounded by m.

(4). Because we still have that $sub(con\mathcal{A}(b)) \subseteq sub(con\mathcal{A}(a))$ if $\{(a,b):r\} \subseteq \mathcal{A}$. So, the depth of each tree in the forest-shaped ABox is bounded by m.(Because the new rule don't add the size of any concepts in A).

Thus, the new Tableau algorithm can always terminate.

Soundness: If the new Tableau algorithm return "consistent", then for $\mathcal{A}$ it is consistent.

And for $\mathcal{T}$, we need prove $\mathcal{T}$ is consistent, Construct the $\mathcal{A}'$ and $\mathcal{I}$ just like the original proof:

1. For any $r \subseteq s$, since $\mathcal{A}'$ is complete, we know for every $\{(a,b):r\} \subseteq \mathcal{A}'$, $\{(a,b):s\} \subseteq \mathcal{A}'$ must hold. Hence, $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ holds.

2. For any disjoint(r,s), $\mathcal{A}'$ is clash-free, so there doesn't exist (a,b) that let $\{(a,b):r\} \subseteq \mathcal{A}'$ and $\{(a,b):s\} \subseteq \mathcal{A}'$ holds at the same time. Hence, $r^{\mathcal{I}} \cap s^{\mathcal{I}} = \emptyset$ holds.

Thus, the new Tableau algorithm can always be sound.

Completeness: If $\mathcal{A}$ is consistent, we should prove that the new two rules application preserves consistency too.

1. $\sqsubseteq -rule$: If $r \subseteq s \in \mathcal{T}$, then $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. So, for $\forall(a,b):r$, $(a,b):s$ always holds, that's mean $\mathcal{A} \cup (a,b):s$ still have the model $\mathcal{I}$. So, $\mathcal{A}$ is still consistent after the rule is applied.

2. If there exists $disjoint(r,s) \in \mathcal{T}$. Since $\mathcal{A}$ is consistent, so there can't be $\{(a,b):r\} \subseteq \mathcal{A}'$ and $\{(a,b):s\} \subseteq \mathcal{A}'$ and the same time. In other words, there is a clash when the condition occurs.

Thus, the new Tableau algorithm can always be complete. Q.E.D

Question 3. 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 $\{A \equiv C \sqcap \neg B, B \equiv P, C \equiv P\}$?

My Solution Question 3. Proof:

Let's denote with $\Delta^{\mathcal{T}}$ and $\Delta^{\mathcal{T}^{\sqsubseteq}}$ the sets of all possible individuals (instances) that can be named by a concept name in $\mathcal{T}$ and $\mathcal{T}^{\sqsubseteq}$, respectively.

First, we will show that if every concept name is satisfiable w.r.t. $\mathcal{T}$, then it is also satisfiable w.r.t. $\mathcal{T}^{\sqsubseteq}$.

Assume that every concept name is satisfiable w.r.t. $\mathcal{T}$. This means that for every concept name A in $\mathcal{T}$, there exists an individual a such that $A(a)$ holds in $\mathcal{T}$. Since $\mathcal{T}^{\sqsubseteq}$ is obtained from $\mathcal{T}$ by replacing each concept definition $A \equiv C$ with the concept inclusion $A \sqsubseteq C$, the satisfaction of $A(a)$ in $\mathcal{T}$ implies the satisfaction of $A(a)$ in $\mathcal{T}^{\sqsubseteq}$. Therefore, every concept name is satisfiable w.r.t. $\mathcal{T}^{\sqsubseteq}$.

Now, we will show that if every concept name is satisfiable w.r.t. $\mathcal{T}^{\sqsubseteq}$, then it is also satisfiable w.r.t. $\mathcal{T}$.

Assume that every concept name is satisfiable w.r.t. $\mathcal{T}^{\sqsubseteq}$. This means that for every concept name A in $\mathcal{T}^{\sqsubseteq}$, there exists an individual a such that $A(a)$ holds in $\mathcal{T}^{\sqsubseteq}$. Since $\mathcal{T}^{\sqsubseteq}$ is a relaxation of $\mathcal{T}$ (by replacing equivalences with subsumptions), the satisfaction of $A(a)$ in $\mathcal{T}^{\sqsubseteq}$ implies the existence of an individual b such that $A(b)$ holds in $\mathcal{T}$. Therefore, every concept name is satisfiable w.r.t. $\mathcal{T}$.

Hence, every concept name is satisfiable w.r.t. $\mathcal{T}$ iff it is satisfiable w.r.t. $\mathcal{T}^{\sqsubseteq}$.

It does not hold.

For the acyclic TBox $\{A \equiv C \sqcap \neg B, B \equiv P, C \equiv P\}$, let's check if every concept name is satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{T}^{\sqsubseteq}$.

In $\mathcal{T}$, we have the following equivalences:

$A \equiv C \sqcap \neg B$

$B \equiv P$

$C \equiv P$

These equivalences form a cycle, which violates the assumption of $\mathcal{T}$ being acyclic. Therefore, this TBox does not satisfy the condition of being acyclic, and the result does not apply to it. Furthermore, the TBox is not in NNF because negation is applied directly to a concept name B, which is not allowed in NNF. ($\mathcal{T}_1$ is not NNF with negation is applied to the defined concept name B)

Question 4. 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:

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

My Solution **Question 4.** Using a metric that should decrease with each transformation, ensuring that the process always terminates to measure the complexity of the $\mathcal{ALC}$ concept.

The metric I choose use is the sum of the number of occurrences of the constructors $\neg$, $\forall$ and $\exists$ in E. I will denote this as $m(E)$.

Let's analyze how this metric changes when applying each transformation:

1. For $\neg(E \sqcap F) \rightsquigarrow \neg\neg\neg E \sqcup \neg\neg\neg F$, $m(E)$ decreases by 2 because two $\neg$ are removed from the concept.
2. For $\neg(E \sqcup F) \rightsquigarrow \neg\neg\neg E \sqcap \neg\neg\neg F$, $m(E)$ decreases by 2 because two $\neg$ are removed from the concept.
3. For $\neg\neg E \rightsquigarrow E$, $m(E)$ decreases by 1 because one $\neg$ is removed from the concept.
4. For $\neg(\exists r.E) \rightsquigarrow \forall r.\neg E$, $m(E)$ increases by 1 because one $\forall$ is added to the concept.
5. For $\neg(\forall r.E) \rightsquigarrow \exists r.\neg E$, $m(E)$ increases by 1 because one $\exists$ is added to the concept.

It is clear that applying transformations (1), (2) and (3) will always decrease $m(E)$, thus making the $\mathcal{ALC}$ concept simpler. On the other hand, transformations (4) and (5) increase $m(E)$, but they also remove a $\neg$ from the concept, which makes it simpler in a different sense.

Therefore, regardless of the order of rule application, exhaustively applying these transformations to an $\mathcal{ALC}$ concept will always terminate, because the complexity of the concept, as measured by $m(E)$, will eventually reach zero.

Question 5. 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.

My Solution **Question 5.** To prove that, we need to show that the algorithm is terminating, sound, and complete.

Termination: The algorithm terminates because it either finds a contradiction (and returns false) or it constructs a model (and returns true). The algorithm terminates in a finite number of steps because:
1. The TBox $\mathcal{T}$ is acyclic, which means there are no infinite chains of role inclusions or disjointness constraints.
2. The ABox $\mathcal{A}$ is finite.
3. Each iteration of the algorithm either adds a new element to the tableau or removes one (by finding a contradiction).
Soundness: The algorithm is sound because if it returns true, then there exists a model for the knowledge base.
Construct a new interpretation $\mathcal{I}'$ based on $\mathcal{I}$. Different from $\mathcal{I}'$, now define that if both $a : A$ and $a : \neg A$ not in $\mathcal{A}'$ but $aI \in CI, \forall A \equiv C \in T$, let $aI' \in AI'$. We firstly prove that I' is also a model of A': In fact, we only need to focus on the condition "C = ¬D" in the proof, as every condition else is consistent with what was discussed in class. And if $C = D$ and $a : C$ in A', then $a : D \in A'$ given that A' is complete. Because A' is clash free, $a : D$ is not in A' and we don't need to use the extended define of I'. Thus, the extension can't cause model I' to be invalid, I' is a model of A' too. For any $A \equiv C$ in T and for any $a : A$ in A', given that A' is complete, $a : C$ in A'. Hence, for $\forall aI' \in AI'$, $aI' \in CI'$ holds. $AI' \subseteq CI'$. And for $\forall aI' \in CI'$ and $a : A \notin A'$, there are two conditions: 1. If $a : A \in A'$, given that A' is complete, $a : C$ in A', there is a clash, where contradiction arises. 2. If $a : A \notin A'$, then $aI' \in CI'$ from the definition. So, for $\forall aI' \in CI'$, then $aI' \in AI'$. $CI' \subseteq AI'$ Therefore, I' satisfies T then satisfies K, which means the Tableau algorithm is sound.
Completeness: The algorithm is complete because if there exists a model for the knowledge base, the algorithm will find it.

If A is consistent, we should prove that the new two rules application preserves consistency too. 1. $\equiv 1-rule$: If $a : A \in A$, $A \equiv C \in T$. So, $aI \in AI = CI$. That means $a : C$ holds too. 2. $\equiv 2 - rule$: If $a : A \in A$, $A \equiv C \in T$. So, $aI \in \Delta I \setminus AI = \Delta I \setminus CI$. That means $a : \neg C$ holds too. Hence, the Tableau algorithm is complete.
In summary, the Tableau algorithm consistent$(\mathcal{T}, \mathcal{A})$ is a decision procedure for the consistency of $\mathcal{ALC}$-knowledge bases with acyclic TBoxes because it is terminating, sound, and complete.

Question 6. 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

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

My Solution **Question 6. No.**
Convert the problem, because the subsumption holds iff. the $\mathcal{T}$ with an ABox $\mathcal{A} = \{a : \neg(\forall r.A) \sqcap \forall r.C, a : \neg(\forall r.E)\}$ is not consistent. Thus, we can apply Tableau Algorithm for ABoxes with Acyclic TBoxes.
First, we transform $\mathcal{A}$ and $\mathcal{T}$ to NNF:
$\mathcal{A} = \{a : \exists r.\neg A \sqcap \forall r.C, a : \exists r.\neg E\}$
$\mathcal{T} = \{C \equiv (\exists r.\neg B) \sqcap \neg A, D \equiv \exists r.B, E \equiv \forall r.\neg A \sqcap \exists r.D\}$ To compute consistent($\mathcal{A}, \mathcal{T}$), we need compute expand($\mathcal{A}$).
($\mathcal{A}$ is not complete now.) First select R = $\sqcap$-rule, α = $(a : \exists r.\neg A \sqcap \forall r.C)$, and we get $\mathcal{A}_1 = \mathcal{A} \cup \{a : \exists r.\neg A, a :$

$\forall r.C\}$
Next, expand($\mathcal{A}_1$): select R = $\exists$-rule, α = $(a : \exists r.\neg E)$, and we get $\mathcal{A}_2 = \mathcal{A}_1 \cup \{(a, b) : r, b : \neg E\}$
Next, expand($\mathcal{A}_2$): select R = $\exists$-rule, α = $(a : \exists r.\neg A)$, and we get $\mathcal{A}_3 = \mathcal{A}_2 \cup \{(a, c) : r, c : \neg A\}$
Next, expand($\mathcal{A}_3$): select R = $\forall$-rule, α = $((a, b) : r, a : \forall r.C)$, and we get $\mathcal{A}_4 = \mathcal{A}_3 \cup \{b : C\}$
Next, expand($\mathcal{A}_4$): select R = $\forall$-rule, α = $((a, c) : r, a : \forall r.C)$, and we get $\mathcal{A}_5 = \mathcal{A}_4 \cup \{c : C\}$
Next, expand($\mathcal{A}_5$): select R = $\equiv$ 1-rule, α = $(b : C, C \equiv (\exists r.\neg B) \sqcap \neg A)$, and we get $\mathcal{A}_6 = \mathcal{A}_5 \cup \{b : (\exists r.\neg B) \sqcap \neg A\}$
Next, expand($\mathcal{A}_6$): select R = $\equiv$ 1-rule, α = $(c : C, C \equiv (\exists r.\neg B) \sqcap \neg A)$, and we get $\mathcal{A}_7 = \mathcal{A}_6 \cup \{c : (\exists r.\neg B) \sqcap \neg A\}$
Next, expand($\mathcal{A}_7$): select R = $\equiv$ 2-rule, α = $(b : \neg E, E \equiv \forall r.\neg A \sqcap \exists r.D)$, and we get $\mathcal{A}_8 = \mathcal{A}_7 \cup \{b : \exists r.A \sqcup \forall r.\neg D\}$
Next, expand($\mathcal{A}_8$): select R = $\sqcap$-rule, α = $(b : (\exists r.\neg B) \sqcap \neg A)$, and we get $\mathcal{A}_9 = \mathcal{A}_8 \cup \{b : \exists r.\neg B, b : \neg A\}$
Next, expand($\mathcal{A}_9$): select R = $\sqcap$-rule, α = $(c : (\exists r.\neg B) \sqcap \neg A)$, and we get $A_{10} = A_9 \cup \{c : \exists r.\neg B, c : \neg A\}$
Next, expand($\mathcal{A}_{10}$): select R = $\exists$-rule, α = $(b : \exists r.\neg B)$, and we get $\mathcal{A}_{11} = \mathcal{A}_{10} \cup \{(b, d) : r, d : \neg B\}$
Next, expand($\mathcal{A}_{11}$): select R = $\exists$-rule, α = $(c : \exists r.\neg B)$, and we get $\mathcal{A}_{12} = \mathcal{A}_{11} \cup \{(c, e) : r, e : \neg B\}$
Next, expand($\mathcal{A}_{12}$): select R = $\sqcup$-rule, α = $(b : \exists r.A \sqcup \forall r.\neg D)$, and we get two choices: $\mathcal{A}_{13,1} = \mathcal{A}_{12} \cup \{b : \exists r.A\}$ or $\mathcal{A}_{13,2} = \mathcal{A}_{12} \cup \{b : \forall r.\neg D\}$
Next, expand($\mathcal{A}_{13,1}$): select R = $\exists$-rule, α = $(b : \exists r.A)$, and we get $\mathcal{A}_{14,1} = \mathcal{A}_{13,1} \cup \{(d, f) : r, f : A\}$.
Now, $\mathcal{A}_{14,1}$ is complete with no clash occurred, so $(\mathcal{T}, \mathcal{A})$ is consistent.
Hence, the subsumption $\neg(\forall r.A) \sqcap \forall r.C \sqsubseteq_{\mathcal{T}} \forall r.E$ **doesn't hold** w.r.t. the acyclic TBox $\mathcal{T}$.

Question 7. 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 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.

My Solution Question 7. Termination: Let's $m = |sub(\mathcal{K})|$.
(1). No rule ever eliminates a statement from $\mathcal{A}$, and every rule application results in adding a new statement of the form $a : C$, where a is an individual name and C is a concept found in sub($\mathcal{A}$) or sub($\mathcal{T}$). Therefore, there can be at most m rule applications that add a concept assertion of the form $a : C$ for any given individual name a, ensuring that $|con_{\mathcal{A}}(a)| \leq m$.

(2). Given that there are no more than m existential restrictions present in $(\mathcal{T}, \mathcal{A})$, an individual name can lead to the addition of at most m new individual names. Consequently, the outdegree of each tree within the forest-like ABox is capped at m (similar to the class proof argument).

(3). As $con_{\mathcal{A}}(a) \subseteq sub(\mathcal{K})$, nodes with less depth in a path of the tree have a younger age. Thus, any path through the tree-structured individuals in the ABox can include at most $2m$ individual names before an individual name b would need to be blocked. This means that the maximum depth of each tree in the forest-shaped ABox is limited by $2m$.

In conclusion, these constraints guarantee that the Tableau algorithm will eventually reach a termination point.

Soundness: Let $\mathcal{A}'$ be the Abox returned by expand($\mathcal{K}$), so $\mathcal{A}'$ is complete. If $\mathcal{A}'$ is clash-free, we should prove that $\mathcal{K}$ is consistent.

Construct a new Abox $\mathcal{A}''$:

$\mathcal{A}'' = \{a : C | a : C \in \mathcal{A}'$ and a is not blocked$\} \cup$

$\{(a, b) : r | (a, b) : r$ and a is not blocked$\} \cup$

$\{(a, b') : r | (a, b) : r$ and a is not blocked and b is blocked by $b'\}$

By definition, it is evident that $\mathcal{A} \subseteq \mathcal{A}''$ since $\mathcal{A} \subseteq \mathcal{A}'$. Furthermore, for any $a : C$ and $(a, b) : r$ present in $\mathcal{A}''$, a and b cannot be blocked. This is due to the fact that all a within $\mathcal{A}''$ remain unblocked. Regarding b, it must originate from the third condition of our definition, which stipulates that there exists a b_0 obstructed by b and b is obstructed by b'. This scenario is impossible because $con_{\mathcal{A}''}(b_0) \subseteq con_{\mathcal{A}''}(b) \subseteq con_{\mathcal{A}''}(b')$ implies there is a sequential path through b_0, b, b', and b would be blocked before we generate b_0. Thus, b must also be blocked.

On the other hand, we can deduce that $con_{\mathcal{A}''}(a) = con_{\mathcal{A}'}(a)$, which leads us to conclude that if $\mathcal{A}'$ is clash-free, then $\mathcal{A}''$ is clash-free as well.

Next, we should prove that $\mathcal{A}'$ is complete implies $\mathcal{A}''$ is complete:

1. If $a : C \sqcap D \in \mathcal{A}''$, then $a : C \sqcap D \in \mathcal{A}'$ because $con_{\mathcal{A}'}(a) = con_{\mathcal{A}''}(a)$. So, $\{a : C, a : D\} \subseteq \mathcal{A}'$, then $\{a : C, a : D\} \subseteq \mathcal{A}''$.
2. If $a : C \sqcup D \in \mathcal{A}''$, then $a : C \sqcup D \in \mathcal{A}'$ because $con_{\mathcal{A}'}(a) = con_{\mathcal{A}''}(a)$. So, $a : C \in \mathcal{A}'$ or $a : D \in \mathcal{A}'$, then $a : C \in \mathcal{A}''$ or $a : D \in \mathcal{A}''$.
3. If $a : C \in \mathcal{A}''$ and $\top \sqsubseteq D \in \mathcal{T}$, then $a : D \in \mathcal{A}'$ because $con_{\mathcal{A}'}(a) = con_{\mathcal{A}''}(a)$. So, $a : D \in \mathcal{A}'$, then $a : D \in \mathcal{A}''$.
4. If $a : \exists r.C \in \mathcal{A}''$ and a is not blocked, then $a : \exists r.C \in \mathcal{A}'$ because $con_{\mathcal{A}'}(a) = con_{\mathcal{A}''}(a)$. So, there is a b that $\{(a, b) : r, b : C\} \subseteq \mathcal{A}'$. Next, we distinguish two cases:

a). If b is not blocked, then $\{(a, b) : r, b : C\} \subseteq \mathcal{A}''$.

b). If b is blocked, given that a is not blocked, from our definition, we know, there is a b' making $(a, b') : r \in \mathcal{A}'$ and $(a, b') : r \in \mathcal{A}''$. And because $con_{\mathcal{A}''}(b) \subseteq con_{\mathcal{A}''}(b')$, we have $b' : C \in \mathcal{A}'$ then $\{(a, b') : r, b' : C\} \subseteq \mathcal{A}''$.

5. If $\{a : \forall r.C, (a, b') : r\} \subseteq \mathcal{A}''$, then $a : \forall r.C \in \mathcal{A}'$ and for whether $(a, b') : r \in \mathcal{A}'$ holds, we distinguish two cases:

a). If $(a, b') : r \in \mathcal{A}'$, then $b' : C \in \mathcal{A}'$. So, $b' : C \in \mathcal{A}''$.

b). If $(a, b') : r \notin \mathcal{A}'$, we can know there is a b blocked by b' in $\mathcal{A}'$. And $(a, b) : r \in \mathcal{A}'$, then $b : C \in \mathcal{A}'$. And because $con_{\mathcal{A}''}(b) \subseteq con_{\mathcal{A}''}(b')$, we get $b' : C \in \mathcal{A}'$. So, $b' : C \in \mathcal{A}''$ holds. **Hence, $\mathcal{A}'$ is complete implies $\mathcal{A}''$ is complete**.

At last, we define a model of $\mathcal{K}$ named $\mathcal{I}$:

$\Delta^{\mathcal{I}} = \{a | a$ is an individual name occurring in $\mathcal{A}''\}$

$a^{\mathcal{I}} = a$ for all individual name a occurring in $\mathcal{A}''$

$A^{\mathcal{I}} = \{a | a \in con_{\mathcal{A}''}(a)\}$

$r^{\mathcal{I}} = \{(a, b) | (a, b) : r \in \mathcal{A}''\}$

And we know it's a model of $\mathcal{A}''$, which proof is just like $\mathcal{ALC}$'s. And for the Tbox $\mathcal{T}$, we assume $\top \sqsubseteq D \in \mathcal{T}$

and for all individual name a in $\mathcal{I}$, we have $a : D$ in $\mathcal{A}''$ because $\mathcal{A}''$ is complete. So, $a^{\mathcal{I}} \in D^{\mathcal{I}}$, which means $\mathcal{T}$ is consistent too.
Because $\mathcal{K}'' = (\mathcal{T}, \mathcal{A}'')$ is consistent, then we know $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent too from previous proof. **Hence, consistent$(\mathcal{T}, \mathcal{A})$ is sound.**
Completeness: Blocking doesn't change anything – it just means that the construction will eventually finish. The only difference lies in the addition of the $\sqsubseteq$-rule: If $a : C \in \mathcal{A}$ and $\top \sqsubseteq D \in \mathcal{T}$, then it must be true that $a^{\mathcal{I}} \in D^{\mathcal{I}}$. Therefore, for any model $\mathcal{I}$ of $(\mathcal{T}, \mathcal{A})$, it is also a model of $(\mathcal{T}, \mathcal{A} \cup (a : D))$. Thus, the expand operation does not alter the consistency of $\mathcal{ALC}$-knowledge bases.
Hence, consistent$(\mathcal{T}, \mathcal{A})$ serves as a decision procedure for the consistency of $\mathcal{ALC}$-knowledge bases with acyclic TBoxes.

Question 8. 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. $\mathcal{T}$.

My Solution Question 8. $\Rightarrow$ If $\mathcal{K}$ is consistent, all $a : C \in \mathcal{A}$ could be satisfied. Therefore, there exists a model $\mathcal{I}$ of $\mathcal{K}$ (which, of course, also models $\{a : C | a : C \in \mathcal{A}\}$ and $\mathcal{T}$), such that for every $a : C \in \mathcal{A}$, it holds that $a^{\mathcal{I}} \in C^{\mathcal{I}}$. Thus, $\mathcal{I}$ is a model of $\mathcal{T}$ that satisfies $a^{\mathcal{I}} \in (C^a_{\mathcal{A}})^{\mathcal{I}}$, which means that the concept description $C^a_{\mathcal{A}}$ is satisfiable w.r.t. T.
$\Leftarrow$ If $C^a_{\mathcal{A}}$ is satisfiable w.r.t. $\mathcal{T}$ for all a occurring in $\mathcal{A}'$, then knowledge base $\mathcal{K}_a = (\mathcal{T}, \mathcal{A}_a)$ is consistent where $\mathcal{A}_a = \{a : C | a : C \in \mathcal{A}'\}$. And we can use Tableau algorithm in $\mathcal{K}_a$ and get a complete and clash-free ABox $\mathcal{A}'_a$ at last. To avoid discussing blocked individual names, we construct $\mathcal{A}''_a$ just like our class based on $\mathcal{A}'_a$ and we know $\mathcal{A}''_a$ is complete and consistent (clash-free):
$\mathcal{A}''_a = \{a : C | a : C \in \mathcal{A}'_a$ and a is not blocked$\} \cup$
$\{(a, b) : r | (a, b) : r \in \mathcal{A}'_a$ and a is not blocked$\} \cup$
$\{(a, b') : r | (a, b) : r \in \mathcal{A}'_a$ and a is not blocked and b is blocked by $b'\}$
Next, we let $\mathcal{A}''$ be a new ABox generated from $\mathcal{A}'$ as follows:

$$\mathcal{A}'' = \left(\bigcup_{a \text{ is an individual name occurring in } \mathcal{A}'} \mathcal{A}''_a \right) \cup \{(a, b) : r \mid (a, b) : r \in \mathcal{A}'$$

It is clear that if $\mathcal{A}''$ is consistent, then $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is consistent because $\mathcal{A} \subseteq \mathcal{A}''$. And if $(\mathcal{T}, \mathcal{A}'')$ is complete and clash-free, then $\mathcal{A}''$ is consistent:
Complete: 1. For $\sqcup$-rule: If $a : C \sqcap D \in \mathcal{A}''$, then there exists an individual name b such that $a : C \sqcap D \in \mathcal{A}''_b$. We know that $\mathcal{A}''_b$ is complete, so $\{a : C, a : D\} \subseteq \mathcal{A}''_b$ and $\{a : C, a : D\} \subseteq \mathcal{A}''$.
2. For $\sqcup$-rule: If $a : C \sqcup D \in \mathcal{A}''$, then there exists an individual name b such that $a : C \sqcup D \in \mathcal{A}''_b$. We know $\mathcal{A}''_b$ is complete, so $a : C \in \mathcal{A}''_b$ or $a : D \in \mathcal{A}''_b$. So, $a : C \in \mathcal{A}''$ or $a : D \in \mathcal{A}''$.
3. For $\forall$-rule: If $a : \forall r.C \in \mathcal{A}''$, $(a, b) : r \in \mathcal{A}''$, discuss two conditions below:
- If $a : D$ occurs in $\mathcal{A}'$, a is a root individual given that be applied all expansion rules except the $\exists$-rule, which only adds concept assertions to ABox. That means $a : \forall r.C \in \mathcal{A}'$, then $b : C \in \mathcal{A}'$, then $b : C \in \mathcal{A}''_b$, and then $b : C \in \mathcal{A}''$.
- If $a : D$ doesn't occur in $\mathcal{A}'$, then a is a tree individual and no rule can add concept assertion with a in fact, which is a contradiction with our precondition. Hence, If $a : \forall r.C \in \mathcal{A}''$, $(a, b) : r \in \mathcal{A}''$, then $b : C \in \mathcal{A}''$.
4. For $\exists$-rule: If $a : \exists r.C \in \mathcal{A}''$, then there exists an individual name b such that $a : \exists r.C \in \mathcal{A}''_b$. Because $\mathcal{A}''_b$ is consistent, then there is an individual name c with $\{(a, c) : r, c : C\} \subseteq \mathcal{A}''_b$. And we know $\{(a, c) : r, c :$

$C\} \subseteq \mathcal{A}''$.
5. For $\sqsubseteq$-rule: If $\top \sqsubseteq C \in \mathcal{T}$, for all individual name a in $\mathcal{A}''$, a must occur in some $\mathcal{A}''_b$ too. Because $\mathcal{A}''_b$ is consistent with $\mathcal{T}$, $a : C \in \mathcal{A}''_b$ and then $a : C \in \mathcal{A}''$.
Clash-free: Because any individual name can't occur in two different $\mathcal{A}''_a$ obviously and any $\mathcal{A}''_a$ is clash-free. Then $\mathcal{A}''$ is clash-free.
Then, $(\mathcal{T}, \mathcal{A}'')$ is complete and clash-free, then $\mathcal{K}$ is complete and clash-free.
Hence, $\mathcal{K}$ is consistent if $C^a_{\mathcal{A}}$ is satisfiable w.r.t. $\mathcal{T}$.
Therefore, $\mathcal{K}$ 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 9. Tableau Algorithm for $\mathcal{ALCN}$

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

My Solution Question 9. Soundness Similar as the proof of $\mathcal{ALC}$, we only need to revise the definition of $\mathcal{A}$":

$$
\begin{aligned}
\mathcal{A}'' = \{&a : C \mid a : C \in \mathcal{A}' \text{ and } a \text{ is not blocked}\}\cup \\
&\{(a,b) : r \mid (a,b) : r \in \mathcal{A}' \text{ and } b \text{ is not blocked}\}\cup \\
&\{(a,bb') : r \mid (a,b) : r \in \mathcal{A}' \text{ and } a \text{ is not blocked and } b \text{ is blocked by } b'\}\cup \\
&\{a = b \mid a = b \in \mathcal{A}' \text{ and } a, b \text{ is not blocked}\}\cup \\
&\{aa' = b \mid a = b \in \mathcal{A}' \text{ and } a \text{ is blocked by } a' \text{ but } b \text{ is not blocked}\}\cup \\
&\{a = b \mid a = b \in \mathcal{A}' \text{ and } a, b \text{ is not blocked}\}\cup \\
&\{aa' = b \mid a = b \in \mathcal{A}' \text{ and } a \text{ is blocked by } a' \text{ but } b \text{ is not blocked}\}\cup \\
&\{a = bb' \mid a = b \in \mathcal{A}' \text{ and } b \text{ is blocked by } b' \text{ but } a \text{ is not blocked}\}\cup \\
&\{aa' = bb' \mid a = b \in \mathcal{A}' \text{ and } a, b \text{ is blocked by } a', b'\}
\end{aligned}
$$

Take care that every time when generating a copy of b' named bb' in the process of generating $\mathcal{A}''$, we need to add the copy to everywhere the individual b' occurs.
If a is not blocked, then bb' is not blocked. (Because all brackets are derived from role assertions, the assignment of the equal sign will not generate a blocked individual either) And since $con_{\mathcal{A}''}(a) = con_{\mathcal{A}''}(a)$ holds too, $\mathcal{A}'$ is clash-free implies $\mathcal{A}''$ is clash-free. Next, we should prove that $\mathcal{A}'$ is complete implies $\mathcal{A}''$ is complete, we only need prove for $\leq$-rule and $\geq$-rule:]
$\leq$-rule: If $a : (\leq nr) \in \mathcal{A}''$, then $a : (\leq nr) \in \mathcal{A}'$ and we can't find distinct $b_0, ..., b_n$ with $\{(a, b_0) : r, (a, b_1) : r, ..., (a, b_n) : r\} \subseteq \mathcal{A}'$. Let's assume that there are distinct $b_0, ..., b_n$ with $\{(a, b_0) : r, (a, b_1) : r, ..., (a, b_n) : r\} \subseteq \mathcal{A}''$. So for these $(a, b_i) : r$ that come from condition $(a, bb') : r$ in fact, we can find the original $(a, b) : r$ in $\mathcal{A}'$ and by this method, we can find $n + 1$ distinct original $(a, b) : r$ correspondingly in $\mathcal{A}'$. This actually creates a contradiction because $\mathcal{A}'$ is complete.
$\geq$-rule: If $a : (\geq nr) \in \mathcal{A}''$ and a is not blocked (In fact, all individual name in $\mathcal{A}''$ is not blocked naturally). We assume there are no distinct $b_1, ..., b_n$ with $\{(a, b_1) : r, ..., (a, b_n) : r\} \subseteq \mathcal{A}''$. Similarly to the proof for $\leq$-rule, we cannot find distinct $b_1, ..., b_n$ with $\{(a, b_1) : r, ..., (a, b_n) : r\} \subseteq \mathcal{A}'$ either. This actually creates a contradiction because A' is complete.
modified$\sqsubseteq$-rule: We only need to prove for the condition "$(b, a) : r \in \mathcal{A}''$, $D \in \mathcal{T}$ and $a : C \in \mathcal{A}''$ (The C can be other concept names except D, eliminating the situation we discussed in class)". We discuss separately:
1. If $(b, a) \in \mathcal{A}'$, $a : D$ in $\mathcal{A}'$ given that $\mathcal{A}'$ is complete. Then $a : D \in \mathcal{A}''$.
2. If $(b, a) \in \mathcal{A}''$, then there is a_0 that is blocked. Then $(b, a) \in \mathcal{A}''$ means a_0 is blocked by a and $(b, a_0) \in \mathcal{A}'$. Since $\mathcal{A}'$ is complete, we know $a_0 : D \in \mathcal{A}'$. Then $a : D \in \mathcal{A}'$ given that $con_{\mathcal{A}'}(a_0) = con_{\mathcal{A}'}(a)$. Because $con_{\mathcal{A}''}(a) = con_{\mathcal{A}'}(a)$, $a : D \in \mathcal{A}''$, which means $\mathcal{A}''$ is complete. And $\mathcal{I}$ with the same definition is a model

of $\mathcal{T}$ and we can prove that just like we do in the class:
$\Delta^{\mathcal{I}} = \{a | a$ is an individual name occurring in $\mathcal{A}''\}$
$a^{\mathcal{I}} = a$ for all individual name a occurring in $\mathcal{A}''$ and $a^{\mathcal{I}} = b^{\mathcal{I}}$ if $a = b \in \mathcal{A}''$
$A^{\mathcal{I}} = \{a | a \in con''_{\mathcal{A}}(a)\}$
$r^{\mathcal{I}} = \{(a,b) | (a,b) : r \in \mathcal{A}''\}$
Based on what we learn in the class, we only need to prove the interpretation $\mathcal{I}$ satisfies $\geq$-rule, $\leq$-rule and modified$\sqsubseteq$-rule:
1. For $\geq$-rule: If $a : C \in \mathcal{A}''$ and $C = \geq nr.$, then completeness of $\mathcal{A}''$ implies that there are distinct $b_1, ..., b_n$ that $\{(a,b_1) : r, (a,b_2) : r, ..., (a,b_n) : r\} \subseteq \mathcal{A}''$. Thus, $\{(a^{\mathcal{I}}, b_1^{\mathcal{I}}), ..., (a^{\mathcal{I}}, b_n^{\mathcal{I}})\} \subseteq r^{\mathcal{I}}$, $a^{\mathcal{I}} \in C^{\mathcal{I}}$ holds.
2. For $\leq$-rule: If $a : C \in \mathcal{A}''$ and $C = \leq nr.$, then completeness of $\mathcal{A}''$ implies that there are no distinct $b_0, b_1, ..., b_n$ that $\{(a,b_0) : r, (a,b_1) : r, ..., (a,b_n) : r\} \subseteq \mathcal{A}''$. Thus, there doesn't exist $\{(a^{\mathcal{I}}, b_0^{\mathcal{I}}), ..., (a^{\mathcal{I}}, b_n^{\mathcal{I}})\} \subseteq r^{\mathcal{I}}$, $a^{\mathcal{I}} \in C^{\mathcal{I}}$ holds.
3. Regarding the modified $\sqsubseteq$-rule: If $D \in \mathcal{T}$, and the fact that all individual names occurring in $\mathcal{A}''$ are listed by the modified rule is accounted for (by appending the condition "$(b,a) : r \in \mathcal{A}$"). Therefore, we are aware that for every individual name a, $a : D$ is present in $\mathcal{A}''$ assuming $\mathcal{A}''$ is comprehensive. Subsequently, we deduce $a^{\mathcal{I}} \in D^{\mathcal{I}}$.
Consequently, $\mathcal{I}$ constitutes a model of $(\mathcal{T}, \mathcal{A}'')$ within the framework of $\mathcal{ALCN}$.
Thus, the soundness of the Tableau algorithm for $\mathcal{ALCN}$ has been substantiated.
Completeness Only need prove completeness for $\geq$-rule, $\leq$-rule and modified$\sqsubseteq$-rule:
1. $\geq$-rule: If $a : (\geq nr)$ and there are no distinct $b_1, ..., b_n$ with $\{(a,b_1) : r, ..., (a,b_n) : r\} \subseteq \mathcal{A}$. If $\mathcal{A}$ is consistent, there are distinct $d_1, ..., d_n$ with $\{(a,d_1) : r, ..., (a,d_n) : r\} \subseteq \mathcal{A}$. So, $\mathcal{A}$'s model is also a model of $\mathcal{A} \cup \{(a,d_1) : r, ..., (a,d_n) : r\} \cup \{d_i = d_j | 1 \leq i < j \leq n\}$.
2. $\leq$-rule: If an entity a is related to distinct entities $b_0, ..., b_n$ with the relation r in a way that $a : (\leq nr)$, as specified by $\{(a,b_1) : r, ..., (a,b_n) : r\} \subseteq \mathcal{A}$, the model of $\mathcal{A}$ guarantees that there cannot be distinct entities $d_1, ..., d_n$ such that $\{(a,d_1) : r, ..., (a,d_n) : r\} \subseteq \mathcal{A}$ exists. Therefore, if $\mathcal{A}$ is consistent, there must exist a pruning operation prune$(A, b_j)[b_j \to b_i] \cup \{b_j = b_i\}$ where $i = j$, and if b_j is a root individual, then b_i will also be a root individual after applying the rule, maintaining consistency. This is because the consistency of $\mathcal{A}$ implies that b_i and b_j exist as individuals in the situation described above. Thus, the knowledge base remains unchanged in its model before and after the application of the rule.
3. modified$\sqsubseteq$-rule: If $(b,a) : r \in \mathcal{A}, T \sqsubseteq \mathcal{A}$ and $a : C \notin \mathcal{A}$ then we also have $a^{\mathcal{I}} \in D^{\mathcal{I}}$.So $\mathcal{I}$ is also a model of $\mathcal{A} \cup \{a : D\}$.
Hence, the Tableau algorithm for $\mathcal{ALCN}$ is complete.

Question 10. 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 $b_1, \dots, b_n$ 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 $d_1, \dots, d_n$ are new individual names

The $\leq$-rule	
Condition:	$\mathcal{A}$ contains $a : (\leq n\, r.C)$, and there are $n+1$ distinct individuals $b_0, \dots, b_n$ with $\{(a,b_i) : r,\ b_i : C \mid 0 \leq i \leq n\} \subseteq \mathcal{A}$
Action:	$\mathcal{A} \longrightarrow$ 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

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

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

My Solution Question 10. The knowledge base is **inconsistent**. And the proposed algorithm **can't detects this**.

1. Because $b^{\mathcal{I}} \in (C \sqcap D)^{\mathcal{I}}$, $b^{\mathcal{I}} \in C^{\mathcal{I}}$, then $b^{\mathcal{I}} \in E^{\mathcal{I}}$, $b^{\mathcal{I}} \in (D \sqcap E)^{\mathcal{I}}$. Given that $a :\leq 1r.(D \sqcap E)$, $b^{\mathcal{I}} = c^{\mathcal{I}}$ holds. But $b^{\mathcal{I}} \in C^{\mathcal{I}}$, $c^{\mathcal{I}} \in (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$. So, $b^{\mathcal{I}}$ is different from $c^{\mathcal{I}}$, this leading to a contradiction. Thus, the knowledge base is not consistent.
2. Transform the knowledge base to NNF:
$(\top \sqsubseteq E \sqcup \neg C, \{a :\leq 1r.(D \sqcap E), (a, b) : r, b : C \sqcap D, (a, c) : r, c : D \sqcap E, c : \neg C\})$
To compute consistent($\mathcal{A}, \mathcal{T}$), we need compute expand($\mathcal{A}$). Firstly, select $R =\sqsubseteq$-rule, $\alpha = (\top \sqsubseteq E \sqcup \neg C, a :\leq 1r.(D \sqcap E), b : C \sqcap D, c : D \sqcap E)$, and we get $\mathcal{A}_1 = \mathcal{A} \cup \{a : E \sqcup \neg C, b : E \sqcup \neg C, c : E \sqcup \neg C\}$
Next, expand($\mathcal{A}_1$): we select $R = \sqcap$-rule, $\alpha = (b : C \sqcap D, c : D \sqcap E)$, and we get $\mathcal{A}_2 = \mathcal{A}_1 \cup \{b : C, b : D, c : D, c : E\}$ Next,expand($\mathcal{A}_2$): select$R = \sqcup$-rule,$\alpha = (b : C \sqcap D, c : D \sqcap E)$,and we can get $2^3 = 8$ choices.
In the 8 conditions, at least $\mathcal{A}_31 = \mathcal{A}_2 \cup \{a : E, b : E, c : E\}$ is complete and clash-free.
So, the proposed algorithm **can't detect** the knowledge base is not consistent.
This is because we cannot obtain $b : D \sqcap E$ with these rules, which might activate the leq-rule that will add $b = c$ to the Abox. This represents the critical juncture at which contradictions are introduced, yet it is absent from the algorithm, hence the proposed algorithm fails to detect whether the knowledge base is inconsistent.

Question 11 (with 1 bonus mark). 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

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

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 $\mathcal{S}$.

My Solution Question 11. To prove the statements, I will go through two items below one by one:
Item 1: $\mathcal{T}$ and $\mathcal{T}' = \{\top \sqsubseteq C_{\mathcal{T}}\}$ have the same models. **Proof:**

Let $\mathcal{I}$ be a model of $\mathcal{T}$. We need to show that $\mathcal{I}$ is also a model of $\mathcal{T}'$.

Since $\mathcal{I}$ is a model of $\mathcal{T}$, for every GCI $C \sqsubseteq D$ in $\mathcal{T}$, we have $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. This implies that $\neg C^{\mathcal{I}} \cup D^{\mathcal{I}} = \top^{\mathcal{I}}$. Therefore, $(\neg C \sqcup D)^{\mathcal{I}} = \top^{\mathcal{I}}$ for every GCI $C \sqsubseteq D$ in $\mathcal{T}$.

Now, consider the concept $C_{\mathcal{T}}$ defined as $\bigsqcap_{C \sqsubseteq D \in \mathcal{T}} \neg C \sqcup D$. Since $(\neg C \sqcup D)^{\mathcal{I}} = \top^{\mathcal{I}}$ for every GCI $C \sqsubseteq D$ in $\mathcal{T}$, it follows that $(C_{\mathcal{T}})^{\mathcal{I}} = \top^{\mathcal{I}}$.

Thus, $\mathcal{I} \models \top \sqsubseteq C_{\mathcal{T}}$, which means $\mathcal{I}$ is also a model of $\mathcal{T}'$.

Conversely, let $\mathcal{I}$ be a model of $\mathcal{T}'$. We need to show that $\mathcal{I}$ is also a model of $\mathcal{T}$.

Since $\mathcal{I}$ is a model of $\mathcal{T}'$, we have $(C_{\mathcal{T}})^{\mathcal{I}} = \top^{\mathcal{I}}$. By the definition of $C_{\mathcal{T}}$, this implies that for every GCI $C \sqsubseteq D$ in $\mathcal{T}$, we have $(\neg C \sqcup D)^{\mathcal{I}} = \top^{\mathcal{I}}$. This further implies that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Therefore, $\mathcal{I}$ satisfies all the GCIs in $\mathcal{T}$, which means $\mathcal{I}$ is a model of $\mathcal{T}$.

Hence, $\mathcal{T}$ and $\mathcal{T}'$ have the same models.

Item 2: Consistency of $\mathcal{K}$ and $\mathcal{K}^+$

Proof:

Need to show two things:

1. If $\mathcal{K}$ is consistent, then $\mathcal{K}^+$ is consistent.
2. If $\mathcal{K}^+$ is inconsistent, then $\mathcal{K}$ is inconsistent.

Firstly, let's assume that $\mathcal{K}$ is consistent. This means that there exists a model I such that $I \models \mathcal{T}$, $I \models \mathcal{A}$, and $I \models \mathcal{R}$. Since $I \models \mathcal{R}$, for every transitivity axiom $\mathsf{trans}(r)$ in $\mathcal{R}$, the interpretation of r in I is transitive.

Now, consider the additional axioms added to $\mathcal{T}^+$. These axioms are of the form $\forall r.C \sqsubseteq \forall r.\forall r.C$ where $\mathsf{trans}(r) \in \mathcal{R}$ and $\forall r.C \in \mathsf{Sub}(C_{\mathcal{T}}) \cup \mathsf{Sub}(A)$. Since I models $\mathcal{T}$ and $\mathcal{A}$, it also models these additional axioms because if r is transitive in I, then $\forall r.C$ implies $\forall r.\forall r.C$. Therefore, I is also a model for $\mathcal{T}^+$. Hence, if $\mathcal{K}$ is consistent, then $\mathcal{K}^+$ is consistent.

Secondly, let's assume that $\mathcal{K}^+$ is inconsistent. This means that there is no model that satisfies all the axioms in $\mathcal{T}^+$, $\mathcal{A}$, and $\mathcal{R}$. However, since $\mathcal{T}^+$ only adds additional axioms based on the transitivity axioms in $\mathcal{R}$, if $\mathcal{K}^+$ is inconsistent, it must be due to the presence of these additional axioms. This implies that there exists at least one transitivity axiom in $\mathcal{R}$ that leads to an inconsistency when combined with the other axioms in $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{R}$. Consequently, $\mathcal{K}$ is also inconsistent because it includes all the axioms from $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{R}$.

Therefore, I have shown that the consistency of $\mathcal{K}$ is equivalent to the consistency of $\mathcal{K}^+$. As a result, the Tableau algorithm for $\mathcal{ALC}$ can be used for reasoning with knowledge bases in $\mathcal{S}$, which extend $\mathcal{ALC}$ with transitivity axioms.