Two Upper Bounds for Conjunctive Query Answering in $\mathcal{SHIQ}$

Carsten Lutz

Institut für Theoretische Informatik
TU Dresden, Germany
lutz@tcs.inf.tu-dresden.de

Abstract. We have shown recently that, in extensions of $\mathcal{ALC}$ that involve inverse roles, conjunctive query answering is harder than satisfiability: it is 2-ExpTime-complete in general and NExpTime-hard if queries are connected and contain at least one answer variable [9]. In this paper, we show that, in $\mathcal{SHIQ}$ without inverse roles (and without transitive roles in the query), conjunctive query answering is only ExpTime-complete and thus not harder than satisfiability. We also show that the NExpTime-lower bound from [9] is tight.

1 Introduction

When description logic (DL) knowledge bases are used in applications with a large amount of instance data, ABox querying is the most important reasoning service. The basic query mechanism for ABoxes is *instance retrieval*, i.e., to return all the individuals from an ABox that are known to be instances of a given query concept. Instance retrieval can be viewed as a well-behaved generalization of subsumption and satisfiability, which are the standard reasoning problems on TBoxes. In particular, algorithms for the latter can typically be adapted to instance retrieval in a straightforward way, and the computational complexity coincides in almost all cases (see [13] for an exception). In 1998, Calvanese et al. introduced *conjunctive query answering* as a more powerful query mechanism for DL ABoxes. Since then, conjunctive queries have received considerable interest in the DL community, see for example the papers [1–4, 6, 7, 11]. In a nutshell, conjunctive query answering generalizes instance retrieval by admitting also queries whose relational structure is not tree-shaped. This generalization is both natural and useful because the relational structure of ABoxes is usually not tree-shaped either.

In contrast to the case of instance retrieval, developing algorithms for conjunctive query answering is not merely a matter of extending algorithms for satisfiability, but requires developing new techniques. In particular, all hitherto known algorithms for DLs that include $\mathcal{ALC}$ as a fragment run in deterministic double exponential runtime, in contrast to algorithms for deciding subsumption and satisfiability which require only exponential time even for DLs much more expressive than $\mathcal{ALC}$. In the recent paper [9], we have shown that this increase in runtime cannot be avoided in extensions of $\mathcal{ALC}$ that include inverse roles. More precisely, we have proved the following two results about $\mathcal{ALCI}$, the extension of $\mathcal{ALC}$ with inverse roles:

(1) Conjunctive query answering in $\mathcal{ALCI}$ is 2-ExpTime-complete.

(2) Rooted conjunctive query answering in $\mathcal{ALCI}$ is co-NExpTime-hard.

Here, *rooted* means that conjunctive queries are required to be connected and contain at least one answer variable. The name "rooted" derives from the fact that every match of such a query is rooted in at least one ABox individual.

This paper is intended as a sequel to [9], providing the upper bounds that have been announced in [9] but not proved in detail. We show that

(3) Conjunctive query answering in $\mathcal{SHQ}$, i.e., $\mathcal{SHIQ}$ without inverse roles, is ExpTime-complete.

(4) Rooted conjunctive query answering in $\mathcal{SHIQ}$ is in co-NExpTime, thus co-NExpTime-complete.

In particular, (3) shows that inverse roles are indeed the culprit for 2-ExpTime-hardness of conjunctive query answering in $\mathcal{ALCI}$ and $\mathcal{SHIQ}$. For both (3) and (4), we assume that non-simple roles (a common generalization of transitive roles in $\mathcal{SHIQ}$) are disallowed in the conjunctive query. We have learned that (3) has been proved independently and in parallel in [12] for the description logic $\mathcal{ALCH}$.

2 Preliminaries

We assume standard notation for the syntax and semantics of $\mathcal{SHIQ}$ knowledge bases [5]. In particular, $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ are countably infinite and disjoint sets of *concept names*, *role names*, and *individual names*. A *TBox* is a set of concept inclusions $C \sqsubseteq D$, role inclusions $r \sqsubseteq s$, and transitivity statements $\mathsf{Trans}(r)$, and a *knowledge base (KB)* is a pair $(\mathcal{T}, \mathcal{A})$ consisting of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. We write $\mathcal{K} \models s \sqsubseteq r$ if the role inclusion $s \sqsubseteq r$ is true in all models of $\mathcal{K}$, and similarly for $\mathcal{K} \models \mathsf{Trans}(r)$. It is easy to see and well-known that "$\mathcal{K} \models s \sqsubseteq r$" and "$\mathcal{K} \models \mathsf{Trans}(r)$" are decidable in polytime [5]. As usual, a role is called *simple* if there is no role s such that $\mathcal{K} \models s \sqsubseteq r$, and $\mathcal{K} \models \mathsf{Trans}(s)$. We write $\mathsf{Ind}(\mathcal{A})$ to denote the set of all individual names in an ABox $\mathcal{A}$. Throughout the paper, the number n inside number restrictions $(\geq n\, r\, C)$ and $(\leq n\, r\, C)$ is assumed to be coded in binary.

Let $\mathsf{N_V}$ be a countably infinite set of *variables*. An *atom* is an expression $C(v)$ or $r(v, v')$, where C is a $\mathcal{SHIQ}$ concept, r is a (possibly inverse) role, and $v, v' \in \mathsf{N_V}$. A *conjunctive query* q is a finite set of atoms. We use $\mathsf{Var}(q)$ to denote the set of variables occurring in the query q. The set $\mathsf{Var}(q)$ is partitioned into *answer variables* and (existentially) *quantified variables*. Let $\mathcal{A}$ be an ABox, $\mathcal{I}$ a model of $\mathcal{A}$, q a conjunctive query, and $\pi : \mathsf{Var}(q) \to \Delta^{\mathcal{I}}$ a total function such that for every answer variable $v \in \mathsf{Var}(q)$, there is an $a \in \mathsf{N_I}$ such that $\pi(v) = a^{\mathcal{I}}$. We write $\mathcal{I} \models^{\pi} C(v)$ if $\pi(v) \in C^{\mathcal{I}}$ and $\mathcal{I} \models^{\pi} r(v, v')$ if $(\pi(v), \pi(v')) \in r^{\mathcal{I}}$. If $\mathcal{I} \models^{\pi} at$ for all $at \in q$, we write $\mathcal{I} \models^{\pi} q$ and call π a *match* for $\mathcal{I}$ and q. We say that $\mathcal{I}$ *satisfies* q and write $\mathcal{I} \models q$ if there is a match π for $\mathcal{I}$ and q. If $\mathcal{I} \models q$ for all models $\mathcal{I}$ of a KB $\mathcal{K}$, we write $\mathcal{K} \models q$ and say that $\mathcal{K}$ *entails* q. The *query entailment problem* is, given a knowledge base $\mathcal{K}$ and a query q, to decide whether $\mathcal{K} \models q$. This is the decision problem corresponding to query answering (which is a search problem), see e.g. [4] for details. Observe that we do not admit the use of individual constants in conjunctive queries. This assumption is only for simplicity, as such constants can easily be simulated by introducing additional concept names [4].

It has been observed many times that, when deciding conjunctive query answering in a DL, it suffices to concentrate on certain regular models of the input knowledge base $\mathcal{K}$. In the following, we describe these models for $\mathcal{SHQ}$. Let $\mathcal{J}$ be an interpretation. A *forest base* $\mathcal{J}$ is an interpretation that interprets transitive roles in an arbitrary way (i.e., not necessarily transitively) and satisfies the following conditions:

(i) $\Delta^{\mathcal{J}}$ is a prefix-closed subset of $\mathbb{N}^+$;
(ii) if $(d, e) \in r^{\mathcal{J}}$, then $e, d \in \mathbb{N}$ or $e = d \cdot c$ for some $c \in \mathbb{N}$.

Elements of $\Delta^{\mathcal{J}} \cap \mathbb{N}$ are the *roots* of $\mathcal{J}$. An interpretation $\mathcal{I}$ is called the $\mathcal{K}$-*closure* of $\mathcal{J}$ if $\mathcal{I}$ is identical to $\mathcal{J}$ except that, for all roles r, we have

$$r^{\mathcal{I}} = r^{\mathcal{J}} \cup \bigcup_{\mathcal{K} \models s \sqsubseteq r \wedge \mathcal{K} \models \mathsf{Trans}(s)} (s^{\mathcal{J}})^+.$$

A model $\mathcal{I}$ of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is a *forest model* of $\mathcal{K}$ if

(iii) $\mathcal{I}$ is the $\mathcal{K}$-closure of a forest base interpretation $\mathcal{J}$, and
(iv) for every root d of $\mathcal{J}$, there is an $a \in \mathsf{Ind}(\mathcal{A})$ such that $a^{\mathcal{I}} = d$.

The *roots* of $\mathcal{I}$ are defined as the roots of $\mathcal{J}$.

Proposition 1. *Let $\mathcal{K}$ be an $\mathcal{SHQ}$-knowledge base and q a conjunctive query. If $\mathcal{K} \not\models q$, then there is a forest model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \not\models q$.*

3 Query Entailment in $\mathcal{SHQ}$ is in ExpTime

We give an algorithm for query entailment in $\mathcal{SHQ}$ that runs in ExpTime, and thus establish ExpTime-completeness of this problem. Our algorithm is inspired by the 2ExpTime algorithm for conjunctive query entailment in $\mathcal{SHIQ}$ with non-simple roles allowed in the query, as given in [4]. On the one hand, our algorithm is simpler because we allow only simple roles in the query. On the other hand, we aim at an ExpTime upper bound which poses additional challenges. In this section, it is convenient to assume that conjunctive queries do not contain answer variables. This assumption can be made w.l.o.g. since answer variables can be simulated using quantified variables and an additional concept name.

The general idea of the algorithm is to (Turing-)reduce query entailment in $\mathcal{SHQ}$ to ABox consistency in $\mathcal{SHQ}^\cap$, i.e., $\mathcal{SHQ}$ extended with role conjunction:

given a $\mathcal{SHQ}$-knowledge base $\mathcal{K}$ and a query q, we produce $\mathcal{SHQ}^\cap$-knowledge bases $\mathcal{K}_1, \ldots, \mathcal{K}_n$ such that $\mathcal{K} \not\models q$ iff any of the $\mathcal{K}_i$ is consisent. This is done such that n is exponential in the size of $\mathcal{K}$ and q, and the size of each knowledge base is polynomial in the size of $\mathcal{K}$ and q. Since knowledge base consistency in $\mathcal{SHQ}^\cap$ can be decided in ExpTime, we obtain the desired ExpTime upper bound for query entailment in $\mathcal{SHQ}$.

Throughout this section, we will sometimes view a conjunctive query as a directed graph $G_q = (V_q, E_q)$ with $V_q = \mathsf{Var}(q)$ and $E_q = \{(v, v') \mid r(v, v') \in q$ for some $r \in \mathsf{N_R}\}$. We call q *tree-shaped* if G_q is a tree. If q is tree-shaped and v is the root of G_q, we call v the root of q.

In the following, we introduce three notions that are central to the construction of the knowledge bases $\mathcal{K}_1, \ldots, \mathcal{K}_n$: fork rewritings, splittings, and spoilers. We start with fork rewritings, and say that

- q' is *obtained from q by fork elimination* if q' is obtained from q by selecting two atoms $r(v', v)$ and $s(v'', v)$ with $v' \neq v''$ and identifying v' and v'';
- q' is a *fork rewriting* of q if q' is obtained from q by repeated (but not necessarily exhaustive) fork elimination;
- q' is a *maximal fork rewriting* of q if q' is a fork rewriting and no further fork elimination is possible in q'.

The following lemma allows us to speak of *the* maximal fork rewriting of a conjunctive query. It is proved in the appendix.

Lemma 1. *Modulo variable renaming, every conjunctive query has a unique maximal fork rewriting.*

The purpose of splittings is to describe such a partition without reference to a concrete model $\mathcal{I}$ and a concrete match π. Let $\mathcal{K}$ be an $\mathcal{SHQ}$-knowledge base. A *splitting* of q w.r.t. $\mathcal{K}$ is a tuple $\Pi = \langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$, where $R, T, S_1, \ldots, S_n$ is a partitioning of $\mathsf{Var}(q)$, $\mu : \{1, \ldots, n\} \to R$ assigns to each set S_i a variable $\mu(i)$ in R, and $\nu : R \to \mathsf{Ind}(\mathcal{A})$ assigns to each variable in R an individual in $\mathcal{A}$. A splitting has to satisfy the following conditions, where $q|_V$ denotes the restriction of q to $V \subseteq \mathsf{Var}(q)$:

(a) some variables v are mapped to a root $\pi(v)$ of $\mathcal{I}$;
(b) other variables v are mapped to a non-root $\pi(v)$ of $\mathcal{I}$, and there is a variable v' such that v is reachable from v' in the directed graph G_q and v' is mapped to a root $\pi(v')$ of $\mathcal{I}$;
(c) yet other variables v are mapped to non-roots $\pi(v)$ of $\mathcal{I}$, but do not satisfy Condition (b).

The purpose of splittings is to describe such a partition without reference to a concrete model $\mathcal{I}$ and a concrete match π. Let $\mathcal{K}$ be an $\mathcal{SHQ}$-knowledge base. A *splitting* of q w.r.t. $\mathcal{K}$ is a tuple $\Pi = \langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$, where $R, T, S_1, \ldots, S_n$ is a partitioning of $\mathsf{Var}(q)$, $\mu : \{1, \ldots, n\} \to R$ assigns to each set S_i a variable $\mu(i)$ in R, and $\nu : R \to \mathsf{Ind}(\mathcal{A})$ assigns to each variable in R an individual in $\mathcal{A}$. A splitting has to satisfy the following conditions, where $q|_V$ denotes the restriction of q to $V \subseteq \mathsf{Var}(q)$:

1. the query $q|_T$ is a variable-disjoint union of tree-shaped queries;
2. the queries $q|_{S_i}$, $1 \leq i \leq n$, are tree-shaped;
3. if $r(v, v') \in q$, then one of the following holds: (i) v, v' belong to the same set $R, T, S_1, \ldots, S_n$ or (ii) $v \in R$, $\mu(i) = v$, and $v' \in S_i$ is the root of $q|_{S_i}$;
4. for $1 \leq i \leq n$, there is an atom $r(\mu(i), v_0) \in q$, with v_0 the root of $q|_{S_i}$.

Intuitively, the R component of a splitting corresponds to Case (a) above, the $S_1, \ldots, S_n$ correspond to Case (b), and T corresponds to Case (c).

Before we introduce spoilers, we establish a central lemma about splittings. We start with some preliminaries. As already noted, we use $\mathcal{SHQ}^\cap$ to denote the extension of $\mathcal{SHQ}$ with a role conjunction operator "$\cap$". However, we restrict the use of role conjunction to *simple* roles. Let q be a tree-shaped conjunctive query. We define a $\mathcal{SHQ}^\cap$-concept $C_{q,v}$ for each variable $v \in \mathsf{Var}(q)$:

– if v is a leaf in G_q, then $C_{q,v} = \bigsqcap_{C(v) \in q} C$;
– otherwise,

$$C_{q,v} = \bigsqcap_{C(v) \in q} C \sqcap \bigsqcap_{(v,v') \in E_q} \exists (\bigcap_{s(v,v') \in q} s).C_{q,v'}).$$

If v is the root of q, we use C_q to abbreviate $C_{q,v}$. The following lemma establishes a connection between forest models and splittings of fork rewritings.

Lemma 2. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base, $\mathcal{I}$ a forest model of $\mathcal{K}$, and q a conjunctive query. Then $\mathcal{I} \models q$ iff there exists a fork rewriting q' of q and a splitting $\langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$ of q' w.r.t. $\mathcal{K}$ such that*

1. *for each disconnected component $\widehat{q}$ of T, there is a $d \in \Delta^\mathcal{I}$ with $d \in (C_{\widehat{q}})^\mathcal{I}$;*
2. *if $C(v) \in q'$ with $v \in R$, then $\nu(v)^\mathcal{I} \in C^\mathcal{I}$;*
3. *if $r(v, v') \in q'$ with $v, v' \in R$, then $(\nu(v)^\mathcal{I}, \nu(v')^\mathcal{I}) \in r^\mathcal{I}$;*
4. *for $1 \leq i \leq n$, we have*

$$\nu(\mu(i))^\mathcal{I} \in \big(\exists (\bigcap_{s(\mu(i), v_0) \in q'} s).C_{q'|_{S_i}}\big)^\mathcal{I}$$

with v_0 root of $q'|_{S_i}$.

Proof. For the "only if" direction, let $\mathcal{I} \models^\pi q$. We construct a fork rewriting q' of q by exhaustively eliminating all forks $r(v', v), s(v'', v) \in q$ with $v' \neq v''$ and $\pi(v') = \pi(v'')$. Now define a splitting $\Pi = \langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$ of q' w.r.t. $\mathcal{K}$ as follows:

– $v \in R$ iff $\pi(v)$ is a root of $\mathcal{I}$; set $\nu(v)$ to some individual name $a \in \mathsf{Ind}(\mathcal{A})$ such that $a^\mathcal{I} = \pi(v)$ (such an a exists by Point (iv) of the definition of forest models);
– $v \in T$ iff there is no variable v' such that $\pi(v')$ is a root of $\mathcal{I}$ and v is reachable from v' in the directed graph $G_{q'}$;
– for each variable v such that $\pi(v)$ is a non-root of $\mathcal{I}$ and there is an $r(v', v) \in q'$ with $\pi(v')$ a root of $\mathcal{I}$, generate a set S_i. The elements of S_i are those variables that are reachable in the directed graph $G_{q'}$ from v. We set $\mu(i) := v'$ (this v' is unique by the construction of q' from q).

It is not hard to see that $R, T, S_1, \ldots, S_n$ is a partition of $\mathsf{Var}(q')$. In particular, the $S_1, \ldots, S_n$ are mutually disjoint and disjoint from R since $\mathcal{I}$ is a forest model. To show that Π is a splitting of q' w.r.t. $\mathcal{K}$, we prove that Conditions 1 to 4 of splittings are satisfied. Conditions 1 and 2 are a consequence of $\mathcal{I}$ being a forest model and the fact that q contains only simple roles. Condition 3 is due to the forest shape of $\mathcal{I}$ and Condition 4 is obvious by construction of Π. It remains to show that Conditions 1 to 4 of Lemma 2 are satisfied. Conditions 2 and 3 are obvious by construction of Π. And finally, Conditions 1 and 4 follow from the construction of Π and the fact that if $\mathcal{I} \models^\pi \widehat{q}$ where $\widehat{q}$ is a tree-shaped query with root v, then $\pi(v) \in (C_{\widehat{q}})^{\mathcal{I}}$.

Conversely, assume that there is a fork rewriting q' of q and a splitting $\langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$ of q' w.r.t. $\mathcal{K}$ such that Conditions 1 to 4 of Lemma 2 are satisfied. Using Conditions 1-4 of splittings and the definition of the concepts $C_{\widehat{q}}$ and $C_{q'|_{S_i}}$, it is straightforward to construct a match π such that $\mathcal{I} \models^\pi q$. The construction starts with setting $\pi(v) = \nu(v)^{\mathcal{I}}$ for all $v \in R$, and then proceeds along edges $r(v, v')$ in the tree-shaped queries $q|_{S_1}, \ldots, q|_{S_n}$ and the tree-shaped components of $q|_T$. ❑

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathcal{SHQ}$-knowledge base, q a conjunctive query, and $\Pi = \langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$ a splitting of q w.r.t. $\mathcal{K}$ such that $q_1, \ldots, q_k$ are the (tree-shaped) disconnected components of $q|_T$. A $\mathcal{SHQ}^\sqcap$-knowledge base $(\mathcal{T}', \mathcal{A}')$ is a *spoiler* for q, $\mathcal{K}$, and Π if one of the following conditions hold:

1. $\top \sqsubseteq \neg C_{q_i} \in \mathcal{T}'$, for some i with $1 \leq i \leq k$;
2. there is an atom $C(v) \in q$ with $v \in R$ and $\neg C(\nu(v)) \in \mathcal{A}'$;
3. there is an atom $r(v, v') \in q$ with $v, v' \in R$ and $\neg r(\nu(v), \nu(v')) \in \mathcal{A}'$;
4. $\neg D(\nu(\mu(i))) \in \mathcal{A}'$ for some i with $1 \leq i \leq n$, and where

$$D = \exists (\bigcap_{s(\mu(i), v_0) \in q} s).C_{q|_{S_i}} \text{ with } v_0 \text{ root of } q|_{S_i}.$$

A $\mathcal{SHQ}^\sqcap$-knowledge base $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ is a *spoiler* for q and $\mathcal{K}$ if (i) for every fork rewriting q' of q and every splitting Π of q' w.r.t. $\mathcal{K}$, $\mathcal{K}'$ is a spoiler for q', $\mathcal{K}$, and Π; and (ii) $\mathcal{K}'$ is minimal with Property (i).

Lemma 3. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathcal{SHQ}$-knowledge base and q a conjunctive query. Then $\mathcal{K} \not\models q$ iff there is a spoiler $(\mathcal{T}', \mathcal{A}')$ for q and $\mathcal{K}$ such that $(\mathcal{T} \cup \mathcal{T}', \mathcal{A} \cup \mathcal{A}')$ is consistent.*

Proof. (sketch) Assume that $\mathcal{K} \not\models q$ and let $\mathcal{I}$ be a model of $\mathcal{K}$ with $\mathcal{I} \not\models q$. By Proposition 1, we may assume that $\mathcal{I}$ is a forest model. We can now assemble a spoiler $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ for q and $\mathcal{K}$ such that $(\mathcal{T} \cup \mathcal{T}', \mathcal{A} \cup \mathcal{A}')$ is consistent by doing the following for every fork rewriting q' of q and every splitting $\Pi = \langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$ of q' w.r.t. $\mathcal{K}$. By Lemma 2, $\mathcal{I} \not\models q$ implies that, for q' and Π, at least one of Conditions 1 to 4 of Lemma 2 are violated. By the corresponding condition from the definition of a spoiler for q', $\mathcal{K}$, and Π, this gives us an axiom to add to $\mathcal{K}'$ such that $\mathcal{K}'$ is a spoiler for q', $\mathcal{K}$, and Π.

Conversely, assume that there is a spoiler $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ for q and $\mathcal{K}$ such that $(\mathcal{T} \cup \mathcal{T}', \mathcal{A} \cup \mathcal{A}')$ is consistent, and let $\mathcal{I}$ be a model of $(\mathcal{T} \cup \mathcal{T}', \mathcal{A} \cup \mathcal{A}')$. We can w.l.o.g. assume that $\mathcal{I}$ is a forest model. Since $\mathcal{I}$ is a model of $\mathcal{K}$, it suffices to show that $\mathcal{I} \not\models q$. Assume to the contrary that $\mathcal{I} \models q$. By Lemma 2, there is a fork rewriting q' of q and a splitting $\Pi = \langle R, T, S_1, \ldots, S_n, \mu, \nu \rangle$ of q' w.r.t. $\mathcal{K}$ such that Conditions 1 to 4 of Lemma 2 are satisfied. By definition of spoilers and since $\mathcal{I}$ is a model of $\mathcal{K}'$, this means that $\mathcal{K}'$ is not a spoiler for q', $\mathcal{K}$, and Π. This is a contradiction to $\mathcal{K}'$ being a spoiler for q and $\mathcal{K}$. ❏

Lemma 3 suggests the following algorithm for deciding conjunctive query entailment in $\mathcal{SHQ}$: given $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and q, enumerate all spoilers $(\mathcal{T}', \mathcal{A}')$ for q and $\mathcal{K}$, return "yes" if for all such spoilers, $(\mathcal{T} \cup \mathcal{T}', \mathcal{A} \cup \mathcal{A}')$ is inconsistent, and "no" otherwise. To prove that this algorithm runs in ExpTime, we have to show that there are at most exponentially many spoilers for q and $\mathcal{K}$ (which can be computed in exponential time), that each spoiler is of size polynomial in the size of q and $\mathcal{K}$, and that consistency of $\mathcal{SHQ}^{\cap}$-knowledge bases can be decided in ExpTime. The latter can be proved by an easy variation of Lemma 6.19 in [14]. It relies on the fact that we restrict the application of role conjunction to simple roles.

Proposition 2. *Consistency of $\mathcal{SHQ}^{\cap}$-KBs is* ExpTime*-complete.*

Now for the number and size of spoilers for q and $\mathcal{K}$. We start with a central lemma about fork rewritings. For a conjunctive query q and $v \in \mathsf{Var}(q)$, let $\mathsf{Reach}_q(v)$ denote the set of all variables in $\mathsf{Var}(q)$ that are reachable from v in the directed graph G_q. Define

$$\mathsf{Trees}(q) := \{q|_{\mathsf{Reach}_q(v)} \mid v \in \mathsf{Var}(q) \text{ and } q|_{\mathsf{Reach}_q(v)} \text{ is tree-shaped}\}.$$

Clearly,

(†) for q^* the maximal fork rewriting of q, the cardinality of $\mathsf{Trees}(q^*)$ is polynomial in the size of q.

Together with the following lemma, this is a crucial observation. The lemma is proved in the appendix.

Lemma 4. *Let q be a conjunctive query, q' a fork rewriting of q, and $\Pi = \langle R, T, S_{v_1}, \ldots, S_{v_n}, \mu, \nu \rangle$ a splitting of q' with $q'_1, \ldots, q'_k$ the disconnected components of $q'|_T$. Moreover, let q^* be the maximal fork rewriting of q. Then $q'_i \in \mathsf{Trees}(q^*)$ for $1 \leq i \leq k$ and $q'|_{S_i} \in \mathsf{Trees}(q^*)$ for $1 \leq i \leq n$.*

Together with (†), the next lemma implies that the size of each spoiler is polynomial in the size of $\mathcal{K}$ and q. We say that a role conjunction $s_1 \cap \cdots \cap s_p$ *occurs* in a conjunctive query q if there are v, v' such that $\{r \mid r(v, v') \in q\} = \{s_1, \ldots, s_p\}$. The proof of the following lemma is based on the direct correspondence between Points 1-4 of Lemma 5 and q Points 1-4 of the definition of spoilers for q and K.

Lemma 5. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathcal{SHQ}$-knowledge base, q a conjunctive query, q^* its maximal fork rewriting, and $\mathcal{K}' = (\mathcal{T}', \mathcal{A}')$ a spoiler for q and $\mathcal{K}$. Then $\mathcal{K}'$ contains only axioms of the following form:*

1. $\top \sqsubseteq \neg C_{q'}$ *with* $q' \in \mathsf{Trees}(q^*)$;
2. $\neg C(a, b)$ *with* $a, b \in \mathsf{Ind}(\mathcal{A})$ *and* C *occurring in* q;
3. $\neg r(a, b)$ *with* $a, b \in \mathsf{Ind}(\mathcal{A})$ *and* r *occurring in* q;
4. $\neg D(a)$ *with* $a \in \mathsf{Ind}(\mathcal{A})$ *and* $D = \exists(s_1 \cap \cdots \cap s_p).C_{q'}$, *where* $s_1 \cap \cdots \cap s_p$ *occurs in* q^* *and* $q' \in \mathsf{Trees}(q^*)$;

It is now easy to establish the intended upper bounds on the number of spoilers. The set of all spoilers can be computed by a straightforward enumeration approach.

Lemma 6. *Let* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *be a* $\mathcal{SHQ}$*-knowledge base and* q *a conjunctive query. Then the number of spoilers for* q *and* $\mathcal{K}$ *is exponential in the size of* q *and* $\mathcal{K}$ *and the set of all spoilers can be computed in time exponential in the size of* q *and* $\mathcal{K}$

Proof. The number of spoilers for q and $\mathcal{K}$ is exponential in the size of q and $\mathcal{K}$ by Lemma 5 and since (i) the cardinality of $\mathsf{Ind}(a)$ is bounded polynomially in the size of $\mathcal{K}$; (ii) the number of role conjunctions occurring in the maximal fork rewriting q^* of q is bounded polynomially in the size of q^* and thus of q; and (iii) by (†), the cardinality of $\mathsf{Trees}(q^*)$ is polynomial in the size of q.

To compute all spoilers in exponential time, we can proceed as follows. Enumerate all exponentially many candidates for spoilers, i.e., all knowledge bases $\mathcal{K}'$ satisfying the conditions listed in Lemma 5. To check whether such a $\mathcal{K}'$ is a spoiler, enumerate all fork rewritings q' of q, of which there are exponentially many in the size of q. Then enumerate all splittings Π of q' w.r.t. $\mathcal{K}$. Their number is polynomial in the size of $\mathcal{K}$ and exponential in the size of q. Finally, it can be checked in time polynomial in q and $\mathcal{K}$ whether $\mathcal{K}'$ is a spoiler for q', $\mathcal{K}$, and Π. If the test succeeds for every q' and Π, $\mathcal{K}'$ is a spoiler for q and $\mathcal{K}$. Otherwise, it is not. ❏

We have thus established the desired ExpTime upper bound. A corresponding lower bound is trivial to show by a reduction of concept satisfiability.

Theorem 1. *Conjunctive query entailment in* $\mathcal{SHQ}$ *is* ExpTime*-complete.*

4 Rooted Query Entailment in $\mathcal{SHIQ}$ is in co-NExpTime

We show that rooted query entailment in $\mathcal{SHIQ}$ is in co-NExpTime. Together with the lower bound in [9], we thus obtain co-NExpTime-completeness.

We start with some preliminaries. A *relaxed forest base* is defined in the same way as a forest base, except that Condition (ii) is relaxed as follows:

(ii)′ if $(d, e) \in r^{\mathcal{J}}$, then $e, d \in \mathbb{N}$ or $e = d \cdot c$ or $d = e \cdot c$ for some $c \in \mathbb{N}$.

A *relaxed forest model* is then defined analogously to a forest model, but based on a relaxed forest base instead of a forest base. The *degree* of a relaxed forest base $\mathcal{I}$ is the maximum cardinality of any set $N \subseteq \mathbb{N}$ such that, for some $d \in \mathbb{N}^*$, we have $\{d \cdot c \mid c \in N\} \subseteq \Delta^{\mathcal{I}}$, and the degree of a relaxed forest model is that of the underlying relaxed forest base. Note that if $\mathcal{I}$ is of degree k, it can have at most k roots. We use $|\mathcal{K}|$ to denote the *size* of the knowledge base $\mathcal{K}$, i.e., the number of symbols needed to write it. Just as for Proposition 3, the proof of the following result is standard.

Proposition 3. *Let $\mathcal{K}$ be a $\mathcal{SHIQ}$-knowledge base and q a conjunctive query. If $\mathcal{K} \not\models q$, then there is a relaxed forest model $\mathcal{I}$ of $\mathcal{K}$ with degree at most $2^{|\mathcal{K}|}$ and such that $\mathcal{I} \not\models q$.*

The main idea of our NExpTime-algorithm for query non-entailment in $\mathcal{SHIQ}$ is as follows. Let $\mathcal{K}$ be a knowledge base and q a conjunctive query. To show that $\mathcal{K} \not\models q$, by Proposition 3 it suffices to find a forest model $\mathcal{I}$ of $\mathcal{K}$ with degree at most $2^{|\mathcal{K}|}$ such that $\mathcal{I} \not\models q$. Now, the crucial observation is that $\mathcal{I} \not\models q$ can be checked by looking only at an "initial part" of $\mathcal{I}$. To see this, assume that $\mathcal{I} \models^{\pi} q$. For $d \in \Delta^{\mathcal{I}}$, the *depth* $||d||$ of d in $\mathcal{I}$ is the length of the word d minus 1. Since we are interested in rooted query containment and $\mathcal{I}$ is a forest model, q has at least one answer variable v and thus $\pi(v)$ is a root of $\mathcal{I}$. Since q is connected and contains only simple roles, $||\pi(u)||$ is bounded by the size $|q|$ of q for every $u \in \mathsf{var}(q)$. It follows that we can check whether $\mathcal{I} \models q$ by looking only at elements $d \in \Delta^{\mathcal{I}}$ with $||d|| \leq |q|$.

Thus, we can decide non-entailment of a query q by a knowledge base $\mathcal{K}$ by guessing an initial part $\mathcal{J}$ of a forest model of $\mathcal{K}$ such that $\mathcal{J}$ is of degree at most $2^{|\mathcal{K}|}$ and of depth at most $|q|$, and then verifying that (i) $\mathcal{J}$ does not match q and (ii) $\mathcal{J}$ can indeed be extended to a full forest model $\mathcal{I}$ of $\mathcal{K}$. When guessing the initial part $\mathcal{J}$, we also have to guess some additional information that is used to ensure (ii). To describe this additional information, we introduce the notion of a type.

W.l.o.g., we assume that all concepts in the input knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ are in NNF, i.e., negation occurs only in front of concept names. For each concept C in NNF, we use $\sim C$ to denote the NNF of $\neg C$, and $\mathsf{cl}_{\mathcal{K}}(C)$ to denote the smallest set such that

1. $C \in \mathsf{cl}_{\mathcal{K}}(C)$,
2. $\mathsf{cl}_{\mathcal{K}}(C)$ is closed under subconcepts and $\sim$, and
3. if $\forall r.D \in \mathsf{cl}_{\mathcal{K}}(C)$, $\mathcal{K} \models \mathsf{Trans}(s)$, and $\mathcal{K} \models s \sqsubseteq r$, then $\forall s.D \in \mathsf{cl}_{\mathcal{K}}(C)$;

Let $C_{\mathcal{T}} = \bigsqcap_{D \sqsubseteq E \in \mathcal{T}} (\sim D \sqcup E)$. The *closure* $\mathsf{cl}(\mathcal{K})$ of $\mathcal{K}$ is defined as $\mathsf{cl}_{\mathcal{K}}(C_{\mathcal{T}}) \cup \bigcup_{C(a) \in \mathcal{A}} \mathsf{cl}_{\mathcal{K}}(C)$.

Definition 1. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base in NNF with $\mathcal{T} = \{\top \sqsubseteq C_{\mathcal{T}}\}$. A* type *for $\mathcal{K}$ is a set $t \subseteq \mathsf{cl}(\mathcal{K})$ such that*

- *$A \in t$ iff $\neg A \notin t$, for all concept names $A \in \mathsf{cl}(\mathcal{K})$;*
- *$C \sqcap D \in t$ implies $C \in t$ and $D \in t$, for all $C \sqcap D \in \mathsf{cl}(\mathcal{K})$;*
- *$C \sqcup D \in t$ implies $C \in t$ or $D \in t$, for all $C \sqcup D \in \mathsf{cl}(\mathcal{K})$;*
- *$\forall r.D \in t$, $\mathcal{K} \models \mathsf{Trans}(s)$, and $\mathcal{K} \models s \sqsubseteq r$ implies $\forall s.D \in t$;*
- *$C_{\mathcal{T}} \in t$.*

We now formalize the initial part of a forest model guessed by the algorithm, which we call a $\mathcal{K},q$-witness. In what follows, the *depth* of a finite forest base $\mathcal{I}$ is the maximum depth of all elements of $\Delta^\mathcal{I}$.

Definition 2. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base in NNF and q a conjunctive query. A $\mathcal{K},q$-*witness* is a pair $(\mathcal{I}, \sigma)$ with $\mathcal{I}$ a finite forest base of degree at most $2^{|\mathcal{K}|}$ and depth at most $|q|$ and σ a mapping that assigns to each $d \in \Delta^\mathcal{I}$ a type $\sigma(d)$ for $\mathcal{K}$ such that for all $d, e \in \Delta^\mathcal{I}$,*

1. *$d \in A^\mathcal{I}$ iff $A \in \sigma(d)$, for all concept names $A \in \mathsf{cl}(\mathcal{K})$;*
2. *if $\exists r.C \in \sigma(d)$ and $||d|| < |q|$, then there is an $e \in \Delta^\mathcal{I}$ such that $(d, e) \in r^\mathcal{I}$ and $C \in \sigma(e)$;*
3. *if $\forall r.C \in \sigma(d)$ and $(d, e) \in r^\mathcal{I}$, then $C \in \sigma(d)$;*
4. *if $\forall r.C \in \sigma(d)$, $(d, e) \in r^\mathcal{I}$, and $\mathcal{K} \models \mathsf{Trans}(r)$, then $\forall r.C \in \sigma(e)$;*
5. *if $(\geqslant n\ r\ C) \in \sigma(d)$ and $||d|| < |q|$, then there are at least n distinct elements $e_1, \ldots, e_n$ such that $(d, e_i) \in r^\mathcal{I}$ and $C \in \sigma(e_i)$ for $1 \leq i \leq n$;*
6. *if $(\leqslant n\ r\ C) \in \sigma(d)$, then there are at most n distinct elements $e_1, \ldots, e_n$ such that $(d, e_i) \in r^\mathcal{I}$ and $C \in \sigma(e_i)$ for $1 \leq i \leq n$;*
7. *if $r(a, b) \in \mathcal{A}$, then $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$;*
8. *if $C(a) \in \mathcal{A}$, then $C \in \sigma(a^\mathcal{I})$;*
9. *if $r \sqsubseteq s \in \mathcal{T}$, then $r^\mathcal{I} \subseteq s^\mathcal{I}$;*
10. *$\mathcal{I} \not\models q$.*

Fix a concept name A_z and a role name r_z not occurring in $\mathcal{K}$. Moreover, let $R_\mathcal{K}$ be the set of all roles occurring in $\mathcal{K}$. For $d \in \Delta^\mathcal{I}$ with $d = e \cdot c$, $c \in \mathbb{N}$, and $||d|| = |q|$, define

$$C_d := \bigsqcap_{C \in \sigma(d)} C \sqcap \exists r_z.(\bigsqcap_{C \in \sigma(e)} C \sqcap A_z) \sqcap \bigsqcap_{r \in R_\mathcal{K} \mid (d,e) \notin r^\mathcal{I}} \forall r.\neg A_z$$

The $\mathcal{K},q$-witness $(\mathcal{I}, \sigma)$ is good *iff for all $d = e \cdot c \in \Delta^\mathcal{I}$ as above, C_d is satisfiable w.r.t. the TBox $\mathcal{T} \cup \{r_z \sqsubseteq r \mid (d, e) \in r^\mathcal{I}\}$.*

Intuitively, a $\mathcal{K},q$-witness being "good" means that it can be extended to a full forest model of $\mathcal{K}$. The use of both d and e in the definition of C_d is similar to the use of double blocking in tableau algorithms for $\mathcal{SHIQ}$, see e.g. [5].

For proving the following lemma, we need the notion of a tree model. An interpretation $\mathcal{J}$ is called *a tree base* if it is a forest base and there is a unique $c \in \mathbb{N}$ such that $c \in \Delta^\mathcal{J}$ (called the *root* of $\mathcal{J}$). A model $\mathcal{I}$ of a concept C is a *tree model* if $\mathcal{I}$ is the $\mathcal{K}$-closure of a tree base $\mathcal{J}$ with root c and $c \in C^\mathcal{I}$.

Lemma 7. *Let $\mathcal{K}$ be a knowledge base in NNF and q a conjunctive query. We have $\mathcal{K} \not\models q$ iff there is a good $\mathcal{K},q$-witness $(\mathcal{I}, q)$.*

Proof. (sketch) For the "only if" direction, assume that $\mathcal{K} \not\models q$. By Proposition 3, there is a forest model $\mathcal{I}$ of $\mathcal{K}$ of degree at most $2^{|\mathcal{K}|}$ such that $\mathcal{I} \not\models q$. Let $\mathcal{J}$ be the forest base underlying $\mathcal{I}$, and let $\mathcal{J}'$ be the restriction of $\mathcal{J}$ to the elements of $\Delta^\mathcal{J}$ that are of depth at most $|q|$. Define a mapping σ, which assigns to each $d \in \Delta^{\mathcal{J}'}$ the set $\sigma(d) := \{C \in \mathsf{cl}^+(\mathcal{K}) \mid d \in C^\mathcal{I}\}$. It can be verified that $(\mathcal{J}', \sigma)$ is a good $\mathcal{K},q$-witness.

For the "if" direction, assume that there is a good $\mathcal{K}, q$-witness $(\mathcal{I}, \sigma)$. Let $d_1, \ldots, d_k$ be the elements in $\Delta^{\mathcal{I}}$ such that $||d_i|| = |q|$. Moreover, let $d_i = e_i \cdot c_i$ for $1 \leq i \leq k$. Since $(\mathcal{I}, \sigma)$ is good, we know that there are models $\mathcal{I}_1, \ldots, \mathcal{I}_k$ for $C_{d_1}, \ldots, C_{d_k}$, respectively. We may w.l.o.g. assume that these models are tree models. For $1 \leq i \leq k$, let r_i be the root of $\mathcal{I}_i$ and let $w_i \in \Delta^{\mathcal{I}_i}$ be such that

$$(r_i, w_i) \in r_z^{\mathcal{I}_i} \text{ and } w_i \in (\bigsqcap_{C \in \sigma(e)} C \sqcap A_z)^{\mathcal{I}_i}.$$

Such w_i exist by definition of C_{d_i}. Modify $\mathcal{I}_1, \ldots, \mathcal{I}_k$ into interpretations $\mathcal{I}'_1, \ldots, \mathcal{I}'_k$ by dropping from each $\mathcal{I}_i$ the root r_i and all elements from $\{w_i\} \cdot \mathbb{N}^*$. Now rename the elements of $\mathcal{I}'_1, \ldots, \mathcal{I}'_k$ such that, for $1 \leq i < j \leq k$, we have $\Delta^{\mathcal{I}'_i} \cap \Delta^{\mathcal{I}} = \emptyset$ and $\Delta^{\mathcal{I}'_i} \cap \Delta^{\mathcal{I}'_j} = \emptyset$. Define a new interpretation $\mathcal{J}$ as the disjoint union of $\mathcal{I}$ and $\mathcal{I}'_1, \ldots, \mathcal{I}'_k$. Next, modify $\mathcal{J}$ into an interpretation $\mathcal{J}'$ by setting

$$\begin{aligned} r^{\mathcal{J}} := r^{\mathcal{J}'} &\cup \{(d_i, d) \mid (r_i, d) \in r^{\mathcal{I}_i} \text{ and } 1 \leq i \leq k\} \\ &\cup \{(d, d_i) \mid (d, r_i) \in r^{\mathcal{I}_i} \text{ and } 1 \leq i \leq k\}. \end{aligned}$$

Finally, let $\mathcal{I}'$ be the $\mathcal{K}$-closure of $\mathcal{J}'$. It is possible to show that $\mathcal{I}'$ is a model of $\mathcal{K}$ and $\mathcal{I}' \not\models q$. ❑

By Lemma 7, the following algorithm decides non-entailment of a query q by a knowledge base $\mathcal{K}$: guess a pair $(\mathcal{I}, \sigma)$ with $\mathcal{I}$ a forest base of degree at most $2^{|\mathcal{K}|}$ and depth at most $|q|$ and σ a mapping that assigns to each $d \in \Delta^{\mathcal{I}}$ a type $\sigma(d)$ for $\mathcal{K}$. Then check whether $(\mathcal{I}, \sigma)$ is a good $\mathcal{K}, q$-witness, return "yes" if it is and "no" otherwise. Since the degree of $\mathcal{I}$ is at most $2^{|\mathcal{K}|}$ and the depth at most $|q|$, the size of $(\mathcal{I}, \sigma)$ is exponential in $|\mathcal{K}| + |q|$. Checking whether $(\mathcal{I}, \sigma)$ is a $\mathcal{K}, q$-witness by verifying Conditions 1-12 from Definition 2 can be done in time exponential in $|\mathcal{K}| + |q|$. Since the concepts C_d are of size polynomial in $|\mathcal{K}| + |q|$ and satisfiability in $\mathcal{SHIQ}$ can be decided in ExpTime, we can check in exponential time whether $(\mathcal{I}, q)$ is good. We thus obtain the following result.

Theorem 2. *Rooted query entailment in $\mathcal{SHIQ}$ is co-*NExpTime*-complete.*

References

1. D. Calvanese, G. De Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proc. of PODS'98*, pages 149–158, 1998.
2. D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of KR'06*, pages 260–270. AAAI Press, 2006.
3. B. Glimm, I. Horrocks, I., and U. Sattler. Conjunctive query entailment for $\mathcal{SHOQ}$. In *Proc. of DL'07*, volume 250 of *CEUR-WS*, 2007.
4. B. Glimm, C. Lutz, I. Horrocks, and U. Sattler. Answering conjunctive queries in the $\mathcal{SHIQ}$ description logic. *JAIR*, 31:150–197, 2008.
5. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proc. of LPAR'99*, number 1705 in LNAI, pages 161–180. Springer, 1999.

6. I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic SHIQ. In *Proc. of CADE-17*, number 1831 in LNCS, pages 482–496. Springer, 2000.
7. U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In *Proc. of IJCAI'05*, pages 466–471. Professional Book Center, 2005.
8. M. Krötzsch, S. Rudolph, and P. Hitzler. Conjunctive queries for a tractable fragment of OWL 1.1. In *Proc. of ISWC'07*, volume 4825 of LNCS, pages 310-323. Springer, 2007.
9. C. Lutz. Inverse roles make conjunctive queries hard. In *Proc. of DL2007*, volume 250 of *CEUR-WS*, 2007.
10. C. Lutz. Two Upper Bounds for Conjunctive Query Answering in (Fragments of) $\mathcal{SHIQ}$. Available from http://lat.inf.tu-dresden.de/∼clu/papers/
11. M. Ortiz, D. Calvanese, and T. Eiter. Characterizing Data Complexity for Conjunctive Query Answering in Expressive Description Logics. In *Proc. of AAAI'07*. AAAI Press, 2007.
12. M. Ortiz, M. Šimkus, and T. Eiter. Worst-case Optimal Conjunctive Query Answering for an Expressive Description Logic without Inverses. In *Proc. of AAAI'08*. AAAI Press, 2008.
13. A. Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *JIIS*, 2:265–278, 1993.
14. S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, 2001.

A Proof of Lemmas 1 and 4

Throughout the appendix, we adopt a special naming scheme for variables in conjunctive queries. Before any fork elimination has taken places, we assume the variables in a conjunctive query to be singleton subsets of a fixed (countably infinite) set Θ. When two variables v and v' are identified during fork elimination, they are both replaced with the variable $v \cup v'$. In this way, each fork elimination carries information about the fork eliminations by which it has been produced (without order). We start with proving two technical lemmas.

Lemma 8. *Let q be a conjunctive query, q' a fork rewriting of q, $v, v' \in \mathsf{Var}(q')$, C a concept, and r a role name. Then*

1. *$C(v) \in q'$ iff there is a $\xi \in v$ with $C(\{\xi\}) \in q$;*
2. *$r(v, v') \in q'$ iff there are $\xi \in v$ and $\xi' \in v'$ with $r(\{\xi\}, \{\xi'\}) \in q$.*

Proof. Straightfoward by induction on the number of fork eliminations that are used during the construction of q' from q. ❏

Lemma 9. *Let q be a conjunctive query, q' a fork rewriting of q, and q^* a maximal fork rewriting of q. Then for every $v \in \mathsf{Var}(q')$, there is a $u \in \mathsf{Var}(q^*)$ with $v \subseteq u$.*

Proof. The proof is by induction on the number of fork eliminations that are used during the construction of q' from q. The induction start (i.e., $q' = q$) is trivial by our naming scheme for variables. For the induction step, let q' be obtained from q'' by eliminating a fork $r(v_1, v_0), s(v_2, v_0) \in q''$. Since the induction hypothesis applies to q'' and all other variables remain unchanged, it suffices to prove that there is a $u \in \mathsf{Var}(q^*)$ with $v_1 \cup v_2 \subseteq u$. By induction hypothesis, there are $u_0, u_1, u_2 \in \mathsf{Var}(q^*)$ with $v_i \subseteq u_i$ for $i \in \{0, 1, 2\}$. It is not hard to see that $r(v_1, v_0), s(v_2, v_0) \in q''$ implies $r(u_1, u_0), s(u_2, u_0) \in q^*$: by Point 2 of Lemma 8, $r(v_1, v_0), s(v_2, v_0) \in q''$ implies that there are $\xi_0, \xi_0', \xi_1, \xi_2 \in \Theta$ such that $r(\{\xi_1\}, \{\xi_0\}), s(\{\xi_2\}, \{\xi_0'\}) \in q$, $\xi_0' \in v_0$, and $\xi_i \in v_i$ for $i \in \{0, 1, 2\}$. Thus $\xi_0' \in u_0$, $\xi_i \in u_i$ for $i \in \{0, 1, 2\}$, and another application of Point 2 of Lemma 8 yields $r(u_1, u_0), s(u_2, u_0) \in q^*$. Since q^* contains no forks, we get $u_1 = u_2$. Clearly, $v_1 \cup v_2 \subseteq u_1 = u_2$ and we are done. ❏

We now come to the proof of Lemmas 1 and 4.

Lemma 1. Modulo variable renaming, every conjunctive query has a unique maximal fork rewriting.

Proof. It suffices to prove that, under our naming scheme, there is a unique maximal fork rewriting of every conjunctive query q. Let q^* and q^{**} be two maximal fork rewritings of q. By Lemma 9, $\mathsf{Var}(q^*) = \mathsf{Var}(q^{**})$. By Lemma 8, we are done. ❏

Lemma 4. Let q be a conjunctive query, q' a fork rewriting of q, and $\Pi = \langle R, T, S_{v_1}, \ldots, S_{v_n}, \mu, \nu \rangle$ a splitting of q' with $q'_1, \ldots, q'_k$ the disconnected components of $q'|_T$. Moreover, let q^* be the maximal fork rewriting of q. Then $q'_i \in \mathsf{Trees}(q^*)$ for $1 \leq i \leq k$ and $q'|_{S_i} \in \mathsf{Trees}(q^*)$ for $1 \leq i \leq n$.

Proof. Let $q_0 = q'$ and $q_0, \ldots, q_m$ be the sequence of queries obtained from q' be exhaustively applying fork elimination. By Lemma 1, $q_m = q^*$. We show by induction on ℓ that the following conditions are satisfied for all $\ell \leq m$, $1 \leq i \leq k$, and $1 \leq j \leq n$:

1. $T \subseteq \mathsf{Var}(q_\ell)$ and $q'|_T = q_\ell|_T$;
2. $q|_T$ is a disconnected component of q_ℓ;
3. $S_j \subseteq \mathsf{Var}(q_\ell)$ and $q'|_{S_j} = q_\ell|_{S_j}$;
4. if $r(v, v') \in q_\ell$ with $\{v, v'\} \cap S_j \neq \emptyset$, then v' is the root of (tree-shaped) $q'|_{S_j}$ and v is the (unique) variable in $\mathsf{Var}(q_\ell)$ with $\mu(j) \subseteq v$.

The induction start is trivial by definition of splittings. For the induction step, we concentrate on Points 3 and 4 (the proof for Points 1 and 2 is similar). Fix an S_j with $1 \leq j \leq n$ and assume that Points 3 and 4 have already been proved for S_j and $q_{\ell-1}$. Let q_ℓ be obtained from $q_{\ell-1}$ by eliminating the fork $r(v', v), s(v'', v) \in q_{\ell-1}$. Since $q_{\ell-1}$ satisfies Point 3 and $q'|_{S_j}$ is tree-shaped by Condition 2 of splittings, none of v, v', v'' is in S_j. It follows that $S_j \subseteq \mathsf{Var}(q_\ell)$ and, by Lemma 8, $q_\ell|_{S_j} = q_{\ell-1}|_{S_j} = q'|_{S_j}$. Since $q_{\ell-1}$ satisfies Point 4, it also follows that q_ℓ satisfies Point 4. This finishes the proof of Points 1 to 4.

It is now simple to show that $q'_i \in \mathsf{Trees}(q^*)$ for $1 \leq i \leq k$ and $q'|_{S_i} \in \mathsf{Trees}(q^*)$ for $1 \leq i \leq n$. Again, we concentrate on the $q'|_{S_i}$. By Point 3, $q'|_{S_i} = q^*|_{S_i}$. It remains to recall that, by Condition 2 of splittings, $q'|_{S_i}$ is tree-shaped. ❏