Conjunctive Query Answering for the Description Logic $\mathcal{SHIQ}$

Birte Glimm[1], Ian Horrocks[1], Carsten Lutz[2], Uli Sattler[1]

[1] School of Computer Science
University of Manchester, UK

[2] Institute for Theoretical Computer Science
TU Dresden, Germany

Abstract

Conjunctive queries play an important role as an expressive query language for Description Logics (DLs). Although modern DLs usually provide for transitive roles, it was an open problem whether conjunctive query answering over DL knowledge bases is decidable if transitive roles are admitted in the query. In this paper, we consider conjunctive queries over knowledge bases formulated in the popular DL $\mathcal{SHIQ}$ and allow transitive roles in both the query and the knowledge base. We show that query answering is decidable and establish the following complexity bounds: regarding combined complexity, we devise a deterministic algorithm for query answering that needs time single exponential in the size of the KB and double exponential in the size of the query. Regarding data complexity, we prove co-NP-completeness.

1 Introduction

Description Logics (DLs) [1] are a well-established family of logic-based knowledge representation formalisms that have recently gained increased attention due to their usage as the logical underpinning of ontology languages such as DAML+OIL and OWL [7]. A DL knowledge base consists of a TBox, which contains terminological knowledge, and an ABox, which contains assertional knowledge. In the TBox, we can define concepts and specify background knowledge. In the ABox, we can use the terms specified in the TBox to describe individuals. Using a database metaphor, the TBox corresponds to the schema, and the ABox corresponds to the data.

Standard DL reasoning services include testing concepts for satisfiability or retrieving instances of a given concept. The latter retrieves all (ABox) individuals that are an instance of the given (possibly complex) concept expression in every model of the knowledge base. The underlying reasoning problems are well-understood, and it is known that the combined complexity of these reasoning problems is ExpTime-complete for $\mathcal{SHIQ}$ [12], where $\mathcal{SHIQ}$ is the DL underlying DAML+OIL and OWL Lite. Despite this high worst-case complexity, efficient implementations of decision procedures for these problems are known. Furthermore, the TBox is usually small compared to the amount of data in the ABox. Therefore, the data complexity of a reasoning problem, i.e., where the complexity is measured in the size of the ABox only, is often a more useful performance estimate. For $\mathcal{SHIQ}$, instance retrievel is known to be data complete for co-NP [9]. However, since instance retrieval only allows for querying the relational structure of the knowledge base in a restricted, tree-like way, it is commonly agreed that a more expressive query language is required, and that conjunctive queries are a suitable basis for this.

To the best of our knowledge, however, no decision procedure is known for conjunctive query answering in $\mathcal{SHIQ}$: the presence of transitive and inverse roles makes the problem rather tricky [4], and results are only available for two kinds of restrictions. The first kind, *grounded* conjunctive queries, is obtained by restricting the variables in queries to be bound to individual names in the ABox only. This results in a form of closed-domain semantics which is different from the usual open-world (and open-domain) semantics in DLs. Motik et al. [10] show that answering *grounded* conjunctive queries for $\mathcal{SHIQ}$ is decidable, and they form the basis for the query language nRQL [5]. In the second kind, the binary atoms in conjunctive queries are restricted to *simple* roles, i.e., to those that are neither transitive nor have transitive sub-roles. For this restriction, decision procedures for various DLs around $\mathcal{SHIQ}$ are known [8, 11], and it is known that answering conjunctive queries is data complete for co-NP [11].

In this paper, we present a decision procedure for conjunctive query answering over $\mathcal{SHIQ}$ knowledge bases without any of these restrictions. We achieve this by transforming the conjunctive query into $\mathcal{SHIQ}^{\sqcap}$-concepts,[1] and showing that conjunctive query answering can be reduced to consistency of $\mathcal{SHIQ}$-knowledge bases extended with $\mathcal{SHIQ}^{\sqcap}$ assertions and GCIs. From our decision procedure, it follows that conjunctive query entailment is data complete for co-NP, and can be decided in time double exponential in the size of the query and single exponential in the size of the knowledge base.

2 Preliminaries

We first introduce the syntax and semantics of $\mathcal{SHIQ}$ and conjunctive queries.

[1] $\mathcal{SHIQ}^{\sqcap}$ is $\mathcal{SHIQ}$ plus role conjunction.

2.1 Syntax and Semantics of $\mathcal{SHIQ}$

Let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$ be sets of *concept names*, *role names*, and *individual names*. We assume that the set $\mathsf{N_R}$ or role names is partitioned into a set $\mathsf{N_{tR}}$ of *transitive role names* and a set $\mathsf{N_{rR}}$ of *normal role names*, i.e., $\mathsf{N_{tR}} \cup \mathsf{N_{rR}} = \mathsf{N_R}$ with $\mathsf{N_{tR}} \cap \mathsf{N_{rR}} = \emptyset$. A *role* is an element of $\mathsf{N_R} \cup \{r^- \mid r \in \mathsf{N_R}\}$, where roles of the form r^- are called *inverse roles*. A *role inclusion* is of the form $r \sqsubseteq s$ with r, s roles. A *role hierarchy* $\mathcal{H}$ is a finite set of role inclusions.

An *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consists of a non-empty set $\Delta^\mathcal{I}$, the *domain* of $\mathcal{I}$, and a function $\cdot^\mathcal{I}$, which maps every concept name A to a subset $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, every role name $r \in \mathsf{N_{rR}}$ to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, every role name $r \in \mathsf{N_{tR}}$ to a transitive binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and every individual name a to an element $a^\mathcal{I} \in \Delta^\mathcal{I}$. An interpretation $\mathcal{I}$ *satisfies* a role inclusion $r \sqsubseteq s$ if $r^\mathcal{I} \subseteq s^\mathcal{I}$ and a role hierarchy $\mathcal{H}$ if it satisfies all role inclusions in $\mathcal{H}$. We use the following standard notation:

1. We define a function Inv which returns the inverse of a role. More precisely, $\mathsf{Inv}(r) := r^-$ if $r \in \mathsf{N_R}$ and $\mathsf{Inv}(r) := s$ if $r = s^-$ for a role name s.

2. Since set inclusion is transitive, we define, for a role hierarchy $\mathcal{H}$, $\sqsubseteq^*_\mathcal{H}$ as the reflexive transitive closure of $\sqsubseteq$ over $\mathcal{H} \cup \{\mathsf{Inv}(r) \sqsubseteq \mathsf{Inv}(s) \mid r \sqsubseteq s \in \mathcal{H}\}$. We use $r \equiv^*_\mathcal{H} s$ as an abbreviation for $r \sqsubseteq^*_\mathcal{H} s$ and $s \sqsubseteq^*_\mathcal{H} r$.

3. For a role hierarchy $\mathcal{H}$ and a role s, we define the set $\mathsf{Trans}_\mathcal{H}$ of transitive roles as $\{s \mid$ there is a role r with $r \equiv s$ and $r \in \mathsf{N_{tR}}$ or $\mathsf{Inv}(r) \in \mathsf{N_{tR}}\}$.

4. A role r is called *simple* w.r.t. a role hierarchy $\mathcal{H}$ if for each role s such that $s \sqsubseteq^*_\mathcal{H} r$, $s \notin \mathsf{Trans}_\mathcal{H}$.

The subscript $\mathcal{H}$ of $\sqsubseteq^*_\mathcal{H}$ and $\mathsf{Trans}_\mathcal{H}$ is dropped if clear from the context. The set of $\mathcal{SHIQ}$*-concepts* (or concepts for short) is the smallest set built inductively from $\mathsf{N_C}$ using the following grammar, where $A \in \mathsf{N_C}$, $n \in \mathbb{N}$, r is a role and s is a simple role:

$$C ::= \top \mid \bot \mid A \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \forall r.C \mid \exists r.C \mid \leqslant ns.C \mid \geqslant ns.C.$$

The semantics of $\mathcal{SHIQ}$-concepts is defined as follows:

$$\begin{array}{rl}
\top^\mathcal{I} = \Delta^\mathcal{I} & (C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I} \qquad (\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I} \\
\bot^\mathcal{I} = \emptyset & (C \sqcup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I} \\
(\forall r.C)^\mathcal{I} = & \{d \in \Delta^\mathcal{I} \mid \text{ if } (d, d') \in r^\mathcal{I}, \text{ then } d' \in C^\mathcal{I}\} \\
(\exists r.C)^\mathcal{I} = & \{d \in \Delta^\mathcal{I} \mid \text{ There is a } (d, d') \in r^\mathcal{I} \text{ with } d' \in C^\mathcal{I}\} \\
(\leqslant ns.C)^\mathcal{I} = & \{d \in \Delta^\mathcal{I} \mid |s^\mathcal{I}(d, C)| \leqslant n\} \\
(\geqslant ns.C)^\mathcal{I} = & \{d \in \Delta^\mathcal{I} \mid |s^\mathcal{I}(d, C)| \geqslant n\}
\end{array}$$

where $|M|$ denotes the cardinality of the set M and $s^{\mathcal{I}}(d, C)$ is defined as

$$\{d' \in \Delta^{\mathcal{I}} \mid (d, d') \in s^{\mathcal{I}} \text{ and } d' \in C^{\mathcal{I}}\}.$$

A *general concept inclusion* (GCI) is an expression $C \sqsubseteq D$, where both C and D are concepts. A finite set of GCIs is called a *TBox.* An interpretation $\mathcal{I}$ *satisfies* a GCI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and a TBox $\mathcal{T}$ if it satisfies each GCI in $\mathcal{T}$. An *assertion* is an expression of the form $C(a), r(a, b)$, $\neg r(a, b)$, or $a \,\dot{\neq}\, b$, where C is a concept, r is a role, $a, b \in \mathsf{N_I}$. An *ABox* is a finite set of assertions. We use $\mathsf{Ind}(\mathcal{A})$ to denote the set of individial names occurring in $\mathcal{A}$, and if $\mathcal{A}$ is clear from the context, we write only Ind. An interpretation $\mathcal{I}$ *satisfies* an assertion $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, $r(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$, $\neg r(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin r^{\mathcal{I}}$, and $a \,\dot{\neq}\, b$ if $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. An interpretation $\mathcal{I}$ *satisfies* an ABox if it satisfies each assertion in $\mathcal{A}$, which we denote with $\mathcal{I} \models \mathcal{A}$.

A *knowledge base* (KB) is a triple ($\mathcal{T}$, $\mathcal{H}$, $\mathcal{A}$) with $\mathcal{T}$ a TBox, $\mathcal{H}$ a role hierarchy, and $\mathcal{A}$ an ABox. Let $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ be a KB and $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ an interpretation. We say that $\mathcal{I}$ *satisfies* $\mathcal{K}$ if $\mathcal{I}$ satisfies $\mathcal{T}$, $\mathcal{H}$, and $\mathcal{A}$. In this case, we say that $\mathcal{I}$ is a *model* of $\mathcal{K}$ and write $\mathcal{I} \models \mathcal{K}$. We say that $\mathcal{K}$ *is consistent* if $\mathcal{K}$ has a model.

2.2 Conjunctive Queries

Now that we have defined the syntax and semantics of $\mathcal{SHIQ}$-concepts and knowledge bases, we are ready to introduce conjunctive queries. Let $\mathsf{N_V}$ be a countably infinite set of *variables* disjoint from $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$. Moreover, let $\mathsf{N_P} = \mathsf{N_C} \cup \mathsf{N_R}$ be the set of *predicate names.*

An *atom* is an expression $A(v)$ or $r(v, v')$, where $A \in \mathsf{N_C}$, r is a role, and $v, v' \in \mathsf{N_V}$. A *conjunctive query* q is a non-empty set of atoms. Intuitively, such a set represents the conjunction of the atoms in the set. We use $\mathsf{Var}(q)$ to denote *the set of variables occurring in* q and we define the *size* $|q|$ of q as the number of atoms in q. Let $\mathcal{I}$ be an interpretation, q a conjunctive query, and $\pi : \mathsf{Var}(q) \to \Delta^{\mathcal{I}}$ a total function. We write

- $\mathcal{I} \models^{\pi} A(v)$ if $(\pi(v)) \in A^{\mathcal{I}}$;
- $\mathcal{I} \models^{\pi} r(v, v')$ if $(\pi(v), \pi(v')) \in r^{\mathcal{I}}$;

If $\mathcal{I} \models^{\pi}$ *at* for all *at* $\in q$, we write $\mathcal{I} \models^{\pi} q$. We say that $\mathcal{I}$ *satisfies* q and write $\mathcal{I} \models q$ if there is a π with $\mathcal{I} \models^{\pi} q$.

One reasoning task regarding conjunctive queries is *query answering.* For introducing this reasoning task, let the variables of a conjunctive query be typed: each variable can either be *non-distinguished*, i.e., existentially quantified or *distinguished.* We call distinguished variables also *answer variables.* Let q be a query in n variables, of which $v_1, \ldots, v_m$ ($m \leq n$) are distinguished. The *answers* of $\mathcal{K}$ to q are those m-tuples $(a_1, \ldots, a_m) \in \mathsf{N_I}^m$ such that for all models $\mathcal{I}$ of $\mathcal{K}$, $\mathcal{I} \models^\pi q$ for some π that satisfies $\pi(v_i) = a_i$ for all i with $1 \leq i \leq m$. Observe that we admit only concept names in atoms $A(v)$, but no complex concepts. This is no restriction since an atom $C(a)$ with C complex can be simulated using the atom $A(a)$ and the concept inclusion $C \sqsubseteq A$.

A reasoning task closely related to query answering is *query entailment*. Here we are given a knowledge base $\mathcal{K}$ and query q and asked whether $\mathcal{I} \models q$ for all models $\mathcal{I}$ of $\mathcal{K}$. If this is the case, we say that $\mathcal{K}$ *entails* q and write $\mathcal{K} \models q$. In this paper, we focus on query entailment. The reasons for this are two-fold: first, query answering can be reduced to query entailment. And second, in contrast to query answering, query entailment is a decision problem and can be studied in terms of classical complexity theory.

We now make the connection between query answering and query entailment more precise. Let $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ be a knowledge base, q a conjunctive query in n variables with answer variables $v_1, \ldots, v_m$, and $t = (a_1, \ldots, a_m) \in \mathsf{N_I}^m$ a tuple. Our aim is to reduce checking whether t is an answer of $\mathcal{K}$ to q to query entailment. To this end, let $\mathcal{A}' := \mathcal{A} \cup \{A_i(a_i) \mid 1 \leq i \leq m\}$, where $A_1, \ldots, A_m$ are concept names that do not occur in $\mathcal{K}$. Moreover, let $\mathcal{K}' := (\mathcal{T}, \mathcal{H}, \mathcal{A}')$ and let $q' := q \cup \{A_i(v_i) \mid 1 \leq i \leq m\}$. The following is not difficult to prove.

Lemma 1. *The tuple t is an answer of $\mathcal{K}$ to q iff $\mathcal{K}'$ entails q'.*

This technique is well known and the newly introduced concept names are often referred to as representative concepts [8] or name formulae [3]. The same technique can be used in order to represent constants (individual names) in the query.

In the rest of this paper, for convenience we assume that conjunctive queries are closed under inverses, i.e., if $r(v, v') \in q$, then $\mathsf{Inv}(r)(v', v) \in q$ and if we add or remove atoms from a query, we implicitly assume that we do this such that the resulting query is again closed under inverses. We will also assume that queries are connected. More precisely, let q be a conjunctive query. We say that q is *connected* if for all $v, v' \in \mathsf{Var}(q)$, there exists a sequence $v_0, \ldots, v_{n-1}$ such that $v_0 = v$, $v_{n-1} = v'$, and for all $i < n-1$, there exists a role r such that $r(v_i, v_{i+1}) \in q$. A collection $q_1, \ldots, q_k$ of queries is a *partitioning* of q if $q = q_1 \cup \cdots \cup q_k$, $\mathsf{Var}(q_i) \cap \mathsf{Var}(q_j) = \emptyset$ for $1 \leq i < j \leq k$, and each q_i is connected. The next lemma says that we can restrict ourselves to the entailment of connected queries. In what follows, we assume queries to be connected without further notice.

Lemma 2. *Let $\mathcal{K}$ be a knowledge base, q a conjunctive query, and $q_1, \ldots, q_n$ a partitioning of q. Then $\mathcal{K} \models q$ iff $\mathcal{K} \models q_i$ for $1 \leq i \leq n$.*

In the rest of this paper, we use q for a connected conjunctive query and $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ for a knowledge base such that, in all assertions $C(a) \in \mathcal{A}$, C is a (possibly negated) concept name. Moreover, for a mapping f, we use $\mathsf{dom}(f)$ and $\mathsf{ran}(f)$ to denote f's domain and range, respectively.

3 Forests and Trees

We will first define canonical (forest-shaped) interpretations, and prove that we can limit our attention to such interpretations.

Definition 3. Let $\mathbb{N}^*$ be the set of all (finite) words over the alphabet $\mathbb{N}$. A *tree* T is a non-empty prefix-closed subset of $\mathbb{N}^*$. For $w, w' \in T$, we call w' a *successor* of w if $w' = w \cdot c$ for some $c \in \mathbb{N}$, where "$\cdot$" denotes concatenation. We call w' a *neighbor* of w if w' is a successor of w or vice versa. Let $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ be a $\mathcal{SHIQ}$ knowledge base. A *forest base for* $\mathcal{K}$ is an interpretation $\mathcal{I}$ that interpretes transitive roles in unrestricted (i.e., not necessarily transitive) relations and additionally satisfies the following conditions:

T1 $\Delta^{\mathcal{I}} \subseteq \mathsf{Ind}(\mathcal{A}) \times \mathbb{N}^*$ such that for all $a \in \mathsf{Ind}(\mathcal{A})$, the set $\{w \mid (a, w) \in \Delta^{\mathcal{I}}\}$ is a tree;

T2 if $((a, w), (a', w')) \in r^{\mathcal{I}}$, then either $w = w' = \varepsilon$ or $a = a'$ and w' is a neighbor of w;

T3 for all $a \in \mathsf{Ind}(\mathcal{A})$, $a^{\mathcal{I}} = (a, \varepsilon)$.

Let $\mathcal{I}$ be an interpretation. Then $\mathcal{I}$ is *canonical for* $\mathcal{K}$ if there exists a forest base $\mathcal{J}$ for $\mathcal{K}$ such that $\mathcal{J}$ is identical to $\mathcal{I}$ except that, for all non-simple roles r, we have

$$r^{\mathcal{I}} = r^{\mathcal{J}} \cup \bigcup_{s \sqsubseteq^* r,\ s \in \mathsf{Trans}} (s^{\mathcal{J}})^+$$

In this case, we say that $\mathcal{J}$ is a forest base *for* $\mathcal{I}$. △

Observe that if $\mathcal{I}$ is canonical for $\mathcal{K}$, then $\Delta^{\mathcal{I}}$ satisfies Condition T1 and T3 above.

Lemma 4. *$\mathcal{K} \not\models q$ iff there exists a canonical model $\mathcal{I}$ with $\mathcal{I} \not\models q$.*

Proof. Using standard unravelling (see e.g. [12]), each model of $\mathcal{K}$ can be converted into a canonical one. Moreover, if $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I}'$ is the canonical model obtained by unravelling $\mathcal{I}$, then it is not hard to show that $\mathcal{I} \not\models q$ implies $\mathcal{I}' \not\models q$, for all conjunctive queries q. ❑

In order to decide whether $\mathcal{K} \models q$, our algorithm will check for the existence of a *counter model*, i.e., a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \not\models q$. Obviously, the above observation means that it suffices to look only for canonical counter models.

Let $\mathcal{I}$ be a canonical model of $\mathcal{K}$, and $\pi : \mathsf{Var}(q) \to \Delta^{\mathcal{I}}$ such that $\mathcal{I} \models^{\pi} q$. We say that π is a *forest match* if for all $r(v, v') \in q$, we have one of the following:

- $\pi(v) = (a, \varepsilon)$ and $\pi(v') = (b, \varepsilon)$ for some $a, b \in \mathsf{Ind}(\mathcal{A})$;
- $\pi(v) = (a, w)$ and $\pi(v') = (a, w')$ for some $a \in \mathsf{Ind}(\mathcal{A})$ and $w, w' \in \mathbb{N}^*$.

Let $\mathcal{I}$ be a canonical model and π a forest match. A variable v is *grounded w.r.t.* $\mathcal{I}$ *and* π if $\pi(v) = (a, \varepsilon)$ for some $a \in \mathsf{Ind}(\mathcal{A})$. A forest match π defines a "partial grounding" for q, and allows us to view q as being split into a set of sub-queries, each of which is mapped into a single tree of $\mathcal{I}$.

We will now describe a series of transformations that we will apply to a query. The first of these, transitive rewriting, will allow us to restrict our attention to forest matches.

Definition 5. Let $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ be a knowledge base and q a conjunctive query. Then a query q' is called a *transitive rewriting* of q w.r.t. $\mathcal{K}$ (or simply a transitive rewriting when $\mathcal{K}$ is obvious from the context) if it is obtained from q by choosing atoms $r_0(v_0, v_0'), \ldots, r_n(v_n, v_n') \in q$ and roles $s_0, \ldots, s_n \in \mathsf{Trans}$ such that $s_i \sqsubseteq^* r_i$ for all $i \leq n$, and then replacing $r_i(v_i, v_i')$ with

$$s_i(v_i, u_i), s_i(u_i, u_i'), s_i(u_i', v_i')$$

or

$$s_i(v_i, u_i), s_i(u_i, v_i')$$

for all $i \leq n$, where u_i and u_i' are variables that do not occur in q. We use $\mathsf{tr}_{\mathcal{K}}(q)$ to denote the set of all transitive rewritings of q w.r.t. $\mathcal{K}$. △

We assume that $\mathsf{tr}_{\mathcal{K}}(q)$ contains no isomorphic queries, i.e., differences in (newly introduced) variable names only are neglected.

Together with Lemma 4, the following lemma shows that in order to decide whether $\mathcal{K}$ entails q, we may enumerate all transitive rewritings q' of q and check whether there is a canonical model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \models^{\pi} q$ with π a forest match.

Lemma 6. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ be a knowledge base, q a conjunctive query, and $\mathcal{I}$ a model of $\mathcal{K}$. Then the following holds:*

1. *If $\mathcal{I}$ is canonical and $\mathcal{I} \models q$, then there is a $q' \in \mathsf{tr}_{\mathcal{K}}(q)$ such that $\mathcal{I} \models^{\pi'} q'$, with π' a forest match.*

2. *If $\mathcal{I} \models q'$ with $q' \in \mathsf{tr}_{\mathcal{K}}(q)$, then $\mathcal{I} \models q$.*

Proof. 1. If there is a forest match π w.r.t. q and $\mathcal{I}$ already, we are done, since q is a transitive rewriting of itself. Therefore, assume that there is no forest match π w.r.t. $\mathcal{I}$ and q. Now, choose any π such that $\mathcal{I} \models^{\pi} q$. Since π is no forest match, there are two variables v, v' with $r(v, v') \in q$ such that $\pi(v) = (a, w), \pi(v') = (a', w'), a \neq a'$, and $w \neq \varepsilon$ or $w' \neq \varepsilon$. We distinguish two cases:

1. Both v and v' are not mapped to roots, i.e., $w \neq \varepsilon$ and $w' \neq \varepsilon$. Since $\mathcal{I} \models^{\pi} r(v, v')$, we have that $(\pi(v), \pi(v')) \in r^{\mathcal{I}}$. Since $\mathcal{I}$ is a canonical (forest) model, there must be a role s with $s \sqsubseteq^{*} r$ and $s \in \mathsf{Trans}_{\mathcal{H}}$ such that $\{(\pi(v), (a, \varepsilon)), ((a, \varepsilon), (a', \varepsilon)), ((a', \varepsilon), \pi(v'))\} \subseteq s^{\mathcal{I}}$. Hence, we can define a transitive rewriting q' of q by replacing $r(v, v')$ with $s(v, u), s(u, u'), s(u', v')$ for u and u' new variables in q. We then define π' as the extension of π that maps u to (a, ε) and u' to (a', ε). It immediately follows that $\mathcal{I} \models^{\pi'} q'$.

2. Either v or v' is mapped to a root. W.l.o.g., let this be v, i.e., $\pi(v) = (a, \varepsilon)$. We can use the same arguments as above: Since $\mathcal{I} \models^{\pi} r(v, v')$, we have that $(\pi(v), \pi(v')) \in r^{\mathcal{I}}$ and since $\mathcal{I}$ is a canonical (forest) model, there must be a role s with $s \sqsubseteq^{*} r$ and $s \in \mathsf{Trans}_{\mathcal{H}}$ such that $\{(\pi(v), (a', \varepsilon)), ((a', \varepsilon), \pi(v'))\} \subseteq s^{\mathcal{I}}$. Hence, we can define a transitive rewriting q' of q by replacing $r(v, v')$ with $s(v, u), s(u, v')$ for $u \notin \mathsf{Var}(q)$ and by defining π' as the extension of π that maps u to (a', ε). It immediately follows that $\mathcal{I} \models^{\pi'} q'$.

We can proceed as described above for each role atom $r(v, v')$ for which $\pi(v) = (a, w)$ and $\pi(v') = (a', w')$ with $a \neq a'$ and $w \neq \varepsilon$ or $w' \neq \varepsilon$. Since each obtained q' is a transitive rewriting of q and $\mathcal{I} \models^{\pi'} q'$ for the extended π', it is clear that we obtain a transitive rewriting q' and a mapping π' such that $\mathcal{I} \models^{\pi'} q'$ and π' is a forest match.

2. If $q = q'$, we are done. Therefore, let $r(v, v') \in q \setminus q'$ and let π be a mapping such that $\mathcal{I} \models^{\pi} q'$. Since q' is a transitive rewriting of q, there is a role s such that $s \sqsubseteq^{*} r$, $s \in \mathsf{Trans}_{\mathcal{H}}$, and either $\{s(v, u), s(u, v')\} \subseteq q'$ or $\{s(v, u), s(u, u'), s(u', v')\} \subseteq q'$. However, since $s \in \mathsf{Trans}_{\mathcal{H}}, (\pi(v), \pi(v')) \in s^{\mathcal{I}}$ and since $s \sqsubseteq^{*} r$, $(\pi(v), \pi(v')) \in r^{\mathcal{I}}$. Therefore, $\mathcal{I} \models^{\pi} q$ as required. ❑

Lemma 7. *Let $\mathcal{K}$ = ($\mathcal{T}$, $\mathcal{H}$, $\mathcal{A}$) be a knowledge base, q a query, $|q| = n$, and $|\mathcal{H}| = m_{\mathcal{H}}$. Then there is a polynomial p such that*

(a) $|\mathsf{tr}_{\mathcal{K}}(q)| \leq 2^{p(n) \cdot \log p(m_{\mathcal{H}})}$

(b) for all $q' \in \mathsf{tr}_{\mathcal{K}}(q)$, $|q'| \leq p(n)$,

Proof. First for (a). Since there are at most n binary atoms in q and at most $m_{\mathcal{H}}$ roles in $\mathsf{Trans}_{\mathcal{H}}$ that may be selected for a transitive rewriting, there are $(m_{\mathcal{H}} + 1)^n = 2^{n \cdot log(m_{\mathcal{H}}+1)}$ such rewritings. For (b), it is easily seen that $|q'| \leq 3n$. ❑

Let q be a query and $\mathcal{I}$ a canonical interpretation. A special case of forest matches are *tree matches*, i.e., matches $\pi : \mathsf{Var}(q) \to \Delta^{\mathcal{I}}$ for which there exists an $a_0 \in \mathsf{Ind}(\mathcal{A})$ such that for all $v \in \mathsf{Var}(q)$, we have $\pi(v) = (a_0, w)$ for some $w \in \mathbb{N}^*$. Intuitively, in this case the whole match concerns only one of the trees in the forest $\Delta^{\mathcal{I}}$, and we call π an *a-tree match* if, for each $v \in \mathsf{Var}(q)$, there is some w such that $\pi(v) = (a, w)$. In our algorithm, forest matches of a query q will be broken down into tree matches of subqueries of q.

We will now show how a query can be rewritten as a tree-shaped query. This procedure, which we call *tree transformation*, can be applied to the subqueries identified by a forest match; we can then use rolling-up to transform each sub-query into a concept.

Tree transformation of q is a three stage process. In the first stage, we derive a *collapsing* q_0 of q by (possibly) identifying variables in q. This allows us, e.g., to transform atoms $r(v, u), r(v, u'), r(u, w), r(u', w)$ into a tree shape by identifying u and u'. In the second stage, we derive an *extension* q_1 of q_0 by (possibly) introducing new variables and role atoms that make redundant existing role atoms $r(v, v')$, where r is non-simple. In the third stage, we derive a *reduct* q' of q_1 by (possibly) removing redundant role atoms, i.e., atoms $r(v, v')$ such that there exist variables $v_0, \ldots, v_n \in \mathsf{Var}(q_1)$ with $v_0 = v$, $v_n = v'$, $s(v_i, v_{i+1}) \in q_1$ for all $i < n$, $s \sqsubseteq^* r$, and $s \in \mathsf{Trans}$. Combining the extension and reduct steps allows us, e.g., to transform a "loop" $r(v, v)$ into a tree shape by introducing a new variable v' and edges $s(v, v'), s(v', v)$ such that $s \sqsubseteq^* r$ and $s \in \mathsf{Trans}$, and then removing the redundant atom $r(v, v)$.

We will now describe this procedure more formally.

Definition 8. Let $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ be a knowledge base. A conjunctive query q is *tree-shaped* if there exists a bijection τ from $\mathsf{Var}(q)$ into a tree such that $r(v, v') \in q$ implies that $\tau(v)$ is a neighbor of $\tau(v')$. Then

- a *collapsing* of q is obtained by identifying variables in q.
- the query q' is an *extension* of q w.r.t. $\mathcal{K}$ if the following hold:
  1. $q \subseteq q'$;
  2. $A(v) \in q'$ implies $A(v) \in q$;
  3. $r(v, v') \in q' \setminus q$ implies that r occurs in $\mathcal{H}$;
  4. $|\mathsf{Var}(q')| \leq 4|q|$;
  5. $|\{r(v, v') \in q' \mid r(v, v') \notin q\}| \leq 171|q|^2$.
- the query q' is a *reduct* of q w.r.t. $\mathcal{K}$ if the following hold:
  1. $q' \subseteq q$;
  2. $A(v) \in q$ implies $A(v) \in q'$;

3. if $r(v, v') \in q \setminus q'$, then there is a role s such that $s \sqsubseteq^* r$, $s \in \mathsf{Trans}$, and there are $v_0, \ldots, v_n$ such that $v_0 = v$, $v_n = v'$, and $s(v_i, v_{i+1}) \in q'$ for all $i < n$.

- a *tree transformations* of q is a query q' for which there are queries q_0 and q_1 such that
  - q_0 is a collapsing of q;
  - q_1 is an extension of q_0 w.r.t. $\mathcal{K}$;
  - q' is a tree-shaped reduct of q_1.

We use $\mathsf{tt}_{\mathcal{K}}(q)$ to denote the set of all tree transformations of q w.r.t. $\mathcal{K}$. △

We note that Condition 5 of extensions is not strictly needed. However, without this condition the algorithm for query entailment to be developed would require double exponential time in the size of the input knowledge base instead of only single exponential time. As in the case of $\mathsf{tr}_{\mathcal{K}}(q)$, we assume that $\mathsf{tt}_{\mathcal{K}}(q)$ does not contain any isomorphic queries.

We now derive an upper bound on the number and size of elements in $\mathsf{tt}_{\mathcal{K}}(q)$. The *size* $|\mathcal{T}|$ ($|\mathcal{H}|, |\mathcal{A}|$) of $\mathcal{T}$ ($\mathcal{H}$, $\mathcal{A}$) is the number of symbols needed to write it. For a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$, the *size* $|\mathcal{K}|$ of $\mathcal{K}$ is the number of symbols needed to write all the components $\mathcal{T}$, $\mathcal{H}$, and $\mathcal{A}$ of $\mathcal{K}$.

Lemma 9. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ be a knowledge base, q a query, $|q| = n$, and $|\mathcal{H}| = m_{\mathcal{H}}$. Then the following hold:*

(a) $|\mathsf{tt}_{\mathcal{K}}(q)| \leq 2^{p(n) \cdot \log p(m_{\mathcal{H}})}$

(b) for all $q' \in \mathsf{tt}_{\mathcal{K}}(q)$, $|q'| \leq p(n)$,

where p is a polynomial.

Proof. (a) is a consequence of the following and some easy computation:

- the number of transitive rewritings of q is bounded by $2^{p'(n) \cdot \log p'(m_{\mathcal{H}})}$ for some polynomial p';
- the number of collapsings of q is bounded by 2^n;
- the number of extensions of q w.r.t. $\mathcal{K}$ is bounded by $3n \cdot (m_{\mathcal{H}} \cdot 8n^2)^{171\, n^2}$;
- the number of reducts of q is bounded by 2^n.

(b) is a consequence of the following:

- if q' is a collapsing of q, then $|q'| \leq n$;
- if q' is an extension of q, then $|q'| \leq 171n^2$;
- if q' is a reduct of q, then $|q'| \leq n$.

❏

Let $\mathcal{K}$ be a knowledge base, q a query, and $q' \in \mathsf{tt}_{\mathcal{K}}(q)$. For each $v \in \mathsf{Var}(q)$, let $\sigma(v)$ be the variable in $\mathsf{Var}(q')$ that v has been identified with ($\sigma(v) = v$ if v has not been identified with another variable). Take mappings $\pi : \mathsf{Var}(q) \to \mathbb{N}^*$ and $\pi' : \mathsf{Var}(q') \to \mathbb{N}^*$. We call π and π' *ε-compatible* iff, for all variables $v \in \mathsf{Var}(q)$, $\pi(v) = \varepsilon$ iff $\pi'(\sigma(v)) = \varepsilon$. Since q' is tree-shaped, π' is a tree with ε as the root and intuitively, ε-compatibility then also guarantees us that we can use $v \in \mathsf{Var}(q)$ for which $\pi(v) = \varepsilon$ as the root or starting point in π and use the above defined transformations in order to transform q into q'.

Lemma 10. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ be a knowledge base, $\mathcal{I}$ a canonical model of $\mathcal{K}$, q a conjunctive query, and π an a-tree match. If $\mathcal{I} \models^{\pi} q$, then there is a $q' \in \mathsf{tt}_{\mathcal{K}}(q)$ and an a-tree match π' such that $\mathcal{I} \models^{\pi'} q'$ and π and π' are ε-compatible.*

Proof. "$\Rightarrow$". Let $\mathcal{J}$ be a forest base of $\mathcal{I}$. Since π is an a-tree match and $\mathcal{I}$ is canonical, we can restrict our attention to the tree rooted in (a, ε) and we call the restriction of $\mathcal{J}$ to domain $(a, \cdot)$ an *a-tree base*. For convenience, we denote the domain elements of an a-tree base with w instead of (a, w).

Let $\mathcal{J}'$ be an a-tree base for $\mathcal{I}$ and assume that $\mathcal{I} \models^{\pi} q$. We now use $\mathcal{I}$, its a-tree base $\mathcal{J}'$, and the mapping π for q to guide the transformation process. Let q_0 be the collapsing of q that is obtained by identifying all variables v, v' with $\pi(v) = \pi(v')$, and let π_0 be the restriction of π to the variables in q_0. It is not hard to verify that $\mathcal{I} \models^{\pi_0} q_0$ and that π_0 is an injection.

Next, we define a query q_1 that is an extension of q_0 and a corresponding mapping π_1. This will involve a number of steps. We start with proving the following:

Claim 1. Let $r(v, v) \in q_0$. Then there exists a neighbor d of $\pi_0(v)$ and a role $s \in \mathsf{Trans}$ such that $s \sqsubseteq^* r$ and $(\pi_0(v), d) \in s^{\mathcal{I}} \cap \mathsf{Inv}(s)^{\mathcal{I}}$.

Proof. Let $r(v, v) \in q_0$. Since $\mathcal{I} \models^{\pi_0} q_0$, we have $(\pi_0(v), \pi_0(v)) \in r^{\mathcal{I}}$. Since $\mathcal{J}'$ is an a-tree base, we have $(\pi_0(v), \pi_0(v)) \notin r^{\mathcal{J}'}$. It follows that there is a sequence $d_0, \ldots, d_{n-1} \in \Delta^{\mathcal{I}}$ and a role $s \in \mathsf{Trans}$ such that $s \sqsubseteq^* r$, $d_0 = \pi_0(v) = d_{n-1}$, and $(d_i, d_{i+1}) \in s^{\mathcal{J}'}$ for $i < n - 1$. Let $d_0, \ldots, d_{n-1}$ be the shortest such sequence. Then it is not hard to see that, because $\mathcal{J}'$ is tree-shaped, we have $d_1 = d_{n-2}$. Since $(d_0, d_1) \in s^{\mathcal{J}'}$ and $(d_{n-2}, d_{n-1}) \in s^{\mathcal{J}'}$ with $d_{n-2} = d_1$ and $d_{n-1} = d_0$, the role s and the element $d = d_1$ are as required. This finishes the proof of Claim 1.

For each $r(v, v) \in q_0$, select a d and s as in Claim 1. We will denote the former with $d_{r,v}$ and the latter with $s_{r,v}$. Now let q^* be obtained from q_0 by doing the following for each $r(v, v) \in q_0$:

- if $d_{r,v} = \pi_0(v')$ for some $v' \in \mathsf{Var}(q_0)$, then add the atoms $s_{r,v}(v, v')$ and $s_{r,v}(v', v)$;
- otherwise, introduce a new variable $v_{r,v}$ and add the atoms $s_{r,v}(v_{r,v}, v)$ and $s_{r,v}(v, v_{r,v})$.

Let π^* be defined as π_0 extended with $\pi^*(v_{r,v}) = d_{r,v}$ for each newly introduced variable $v_{r,v}$. It is easily seen that q^* is connected, π^* is injective, and $\mathcal{I} \models^{\pi^*} q^*$.

For $w, w' \in \mathbb{N}^*$, the *longest common prefix (LCP)* of w, w' is the longest $w^* \in \mathbb{N}^*$ such that w^* is prefix of both w and w'. Set

$$D := \mathsf{ran}(\pi^*) \cup \{d \in \Delta^{\mathcal{I}} \mid \exists v, v' \in \mathsf{Var}(q') : d \text{ LCP of } \pi^*(v), \pi^*(v')\}.$$

The following are not too difficult to see:

Fact (a) if $d, d' \in D$, then the LCP of d, d' is in D;

Fact (b) $|D| \leq 2|\mathsf{Var}(q^*)|$ (since $2n$ is an upper bound on the number of nodes in a tree that has n leaves and is at least binarily branching at every non-leaf).

Set $V := \mathsf{Var}(q^*) \cup \{v_d \mid d \in D \setminus \mathsf{ran}(\pi^*)\}$ and $\pi_1 := \pi^* \cup \{v_d \mapsto d \mid v_d \in V\}$. The members of V will be the variables of the new query q_1.

Next, we prove a technical claim. Let $d, d' \in \Delta^{\mathcal{I}}$. The *path* from d to d' is the (unique) shortest sequence of elements $d_0, \ldots, d_{n-1} \in \Delta^{\mathcal{I}}$ such that $d_0 = d$, $d_{n-1} = d'$, and d_{i+1} is a neighbor of d_i for all $i < n-1$. The *length of a path* is the number of elements in it, i.e., the path $d_0, \ldots, d_{n-1}$ is of length n.

Claim 2. Let $d \in D \setminus \mathsf{ran}(\pi^*)$. Then there is a $v \in \mathsf{Var}(q^*)$ and a role s such that $(d, \pi^*(v)) \in s^{\mathcal{I}}$.

Proof. Since $d \in D \setminus \mathsf{ran}(\pi^*)$, there are $v, v' \in \mathsf{Var}(q^*)$ such that d is LCP of $\pi^*(v)$ and $\pi^*(v')$. Since q^* is connected, there is a sequence $v_0, \ldots, v_{n-1} \in \mathsf{Var}(q^*)$ such that $v_0 = v$, $v_{n-1} = v'$, and for all $i < n-1$, there is a role r_i such that $r_i(v_i, v_{i+1}) \in q^*$. We distinguish two cases:

1. There is an $i < n$ such that d is not a prefix of $\pi^*(v_i)$.

   Since d is a prefix of $\pi^*(v_0)$ and $\pi^*(v_{n-1})$, it follows that there is an $\ell < n-1$ such that d is a prefix of $\pi^*(v_{\ell+1})$, but not of $\pi^*(v_\ell)$. Let $d_0, \ldots, d_{m-1}$ be the path from $\pi^*(v_\ell)$ to $\pi^*(v_{\ell+1})$. Let $k < m-1$ such that $d_k = d$ (such a k clearly exists). Since $r_\ell(v_\ell, v_{\ell+1}) \in q^*$ and $\mathcal{I} \models^{\pi^*} q^*$, we have $(d_0, d_{m-1}) = (\pi^*(v_l), \pi^*(v_{l+1})) \in r_\ell^{\mathcal{I}}$. We have three subcases:

   - $m = 1$. Impossible since $\pi^*(v_\ell) \neq \pi^*(v_\ell + 1)$.
   - $m = 2$. Then d_0 and d_{m-1} are neighbors. Since d is a prefix of $\pi^*(v_{\ell+1})$ but not of $\pi^*(v_\ell)$, we have $\pi^*(v_\ell) = d$. Contradiction to $d \notin \mathsf{ran}(\pi^*)$.
   - $m > 2$. Then $(d_0, d_{m-1}) \in r_\ell^{\mathcal{I}} \setminus r_\ell^{\mathcal{J}'}$. By construction of $\mathcal{I}$ from $\mathcal{J}'$, this means that there is a role $s \in \mathsf{Trans}$ such that $s \sqsubseteq^* r_\ell$ and $(d_i, d_{i+1}) \in s^{\mathcal{J}'}$ for all $i < m-1$. Again by construction of $\mathcal{I}$, this means $(d_k, d_{m-1}) \in s^{\mathcal{I}}$. Since $\pi^*(v_{\ell+1}) = d_{m-1}$, we have $(d, \pi^*(v_{\ell+1})) \in s^{\mathcal{I}}$ and are done.

2. For all $i < n$, d is a prefix of $\pi(v_i)$.

   Since d is LCP of $\pi^*(v)$ and $\pi^*(v')$ and $d \notin \mathsf{ran}(\pi^*)$, d is distinct from $\pi^*(v)$ and $\pi^*(v')$ and we have $\pi^*(v) = d \cdot w$ and $\pi^*(v') = d \cdot w'$ for some $w, w' \in \mathbb{N}^*$ such that the first symbol of w is different from the first symbol of w' ($\pi^*(v)$ and $\pi^*(v')$ are in different branches of the tree). Since d is a prefix of $\pi(v_i)$ for all $i < n$, it follows that there is an $\ell < n - 1$ such that $\pi^*(v_\ell)$ and $\pi^*(v_{\ell+1})$ are in different branches of the tree, i.e., $\pi^*(v_\ell) = d \cdot u$ and $\pi^*(v_{\ell+1}) = d \cdot u'$ for some $u, u' \in \mathbb{N}^*$ such that the first symbol of u is different from the first symbol of u'.

   Since $(\pi^*(v_\ell), \pi^*(v_{\ell+1})) \in r_\ell^{\mathcal{I}}$, $\mathcal{I}$ is a canonical interpretation based on $\mathcal{J}$, and $\mathsf{ran}(\pi^*)$ is restricted to the a-tree base $\mathcal{J}'$, this means that there is an $s \in \mathsf{Trans}$ such that

   - $s \sqsubseteq^* r_\ell$,
   - $(d \cdot \hat{u} \cdot c, d \cdot \hat{u}) \in s^{\mathcal{J}'}$ for each prefix $\hat{u} \cdot c$ of u (with $u \in \mathbb{N}^*$ and $c \in \mathbb{N}$),
   - and $(d \cdot \hat{u}', d \cdot \hat{u}' \cdot c) \in s^{\mathcal{J}'}$ for each prefix $\hat{u}' \cdot c$ of u'.

   Since $s \in \mathsf{Trans}$, it follows that $(d, \pi^*(v_{\ell+1})) \in s^{\mathcal{I}}$ and we are done.

This finishes the proof of Claim 2.

Let q^{**} be obtained from q^* as follows: for each $d \in D \backslash \mathsf{ran}(\pi^*)$, select a variable v and role s as in Claim 2 and include $s(v_d, v)$ in q^{**}. The query q^{**} is our starting point for defining q_1, which will contain additional atoms (but no additional variables). It is obvious that q^{**} is connected and that $\mathcal{I} \models^{\pi_1} q^{**}$.

Next, we prove another technical claim. Let $d, d' \in \Delta^{\mathcal{I}}$ and $d_0, \ldots, d_{n-1}$ with $d_0 = d$ and $d_{n-1} = d'$ the path from d to d'. The *relevant path* $d'_0, \ldots, d'_{m-1}$ from d to d' is the subsequence of $d_0, \ldots, d_{n-1}$ that is obtained by dropping all elements not in the range of π_1.

Claim 3. Let $r(v, v') \in q^{**}$ such that the length n of the relevant path $d_0, \ldots, d_{n-1}$ from $\pi_1(v)$ to $\pi_1(v')$ is greater than 2. Then there exists an $s \in \mathsf{Trans}$ such that $s \sqsubseteq^* r$ and $(d_i, d_{i+1}) \in s^{\mathcal{I}}$ for all $i < n - 1$.

Proof. Let $d'_0, \ldots, d'_{m-1}$ be the path from $\pi_1(v)$ to $\pi_1(v')$. Then $n > 2$ implies $m > 2$. We have to show that there is a role s as in the claim. Since $\mathcal{I} \models^{\pi_1} q^{**}$, $m > 2$ implies $(\pi(v), \pi(v')) \in r^{\mathcal{I}} \setminus r^{\mathcal{J}}$. Since $\mathcal{I}$ is based on $\mathcal{J}$, it follows that there is an $s \in \mathsf{Trans}$ such that $s \sqsubseteq^* r$, and $(d'_i, d'_{i+1}) \in s^{\mathcal{J}}$ for all $i < m - 1$. By construction of $\mathcal{I}$ from $\mathcal{J}$, it follows that $(d_i, d_{i+1}) \in s^{\mathcal{I}}$ for all $i < n - 1$, which finishes the proof of the claim.

Now let q_1 be obtained from q^{**} as follows: for all atoms $r(v, v') \in q^{**}$ with relevant path $d_0, \ldots, d_{n-1}$ such that $n > 2$, select a role s as in Claim 3, include $s(v_0, v_1), \ldots, s(v_{n-2}, v_{n-1})$ in q_1, where $v_0, \ldots, v_{n-1} \in V$ are the variables such that $v_i = d_i$ for all $i < n$ (there is no ambiguity here since π_1 is injective). It is obvious that $\mathcal{I} \models^{\pi_1} q_1$.

Let us show that q_1 is an extension of q^*. Conditions 1 to 3 are easily verified. For Condition 4, note that q^* contains at most $|q_0| \leq |q|$ additional nodes and q^{**} contains at most $|q^*|$ additional nodes by Fact (b) above.

For Condition 5, we note the following:

- The number of atoms in $q^* \setminus q_0$ is bounded by $4|q_0|$ (4 instead of 2 because queries are closed under inverse roles).

- The number of atoms in $q^{**} \setminus q^*$ is bounded by $4|q_0|$ (we add at most one edge for each of the $2|q_0|$ nodes in q^* and close off under inverse roles).

- The number of atoms in $q_1 \setminus q^{**}$ is bounded by $2|q^{**}|^2$. This is due to the following facts: There are at most $|q^{**}|$ binary atoms in q^*, each gives rise to a single relevant path, every relevant path has length at most $|q^{**}|$ and in the worst case we introduce a new binary atom (plus its inverse) for each edge of each such relevant path. Therefore, we can use $2|q^{**}|^2$ as a bound for the number of new role atoms in q_1.

- Since $|q^*| \leqslant |q_0|+4|q_0| = 5|q_0|$ and $|q^{**}| \leqslant |q^*|+4|q_0| = 9|q_0|$, the size of new atoms in $q_1 \setminus q^{**}$ is bounded by $2(9|q_0|)^2 = 162|q_0|^2$ and, overall, the number of new role atoms in q_1 (compared to q_0) is bounded by $162|q_0|^2 + 9|q_0| \leqslant 171|q_0|^2$.

Now, let q' be the query obtained from q_1 by dropping all $r(v, v') \in q_1$ such that $v = v'$ or the relevant path from $\pi_1(v)$ to $\pi_1(v')$ is longer than two elements. Also, let $\pi' = \pi_1$. Clearly, $\mathcal{I} \models^{\pi'} q'$ and π' is injective. We show that q' is a reduct of q_1. As the first two conditions of reducts are easily verified, we concentrate on the third. Assume that $r(v, v') \in q_1 \setminus q'$. There are two cases:

- $v = v'$. It can be checked that in the process described above, we do not introduce new role atoms $s(u, u')$ such that $u = u'$, i.e., $s(u, u') \in (q^* \setminus q_0) \cup (q^{**} \setminus q^*)$ implies $u \neq u'$, and thus we have $r(v, v) \in q_0$. By construction of q^*, there is a $v^* \in \mathsf{Var}(q^*)$ and an $s \in \mathsf{Trans}$ such that $v \neq v^*$, $s \sqsubseteq^* r$, $s(v, v^*) \in q^*$, $s(v^*, v) \in q^*$, and $\pi^*(v)$ is a neighbor of $\pi^*(v^*)$. By building a reduct we only drop role atoms $r(v, v')$ with $v = v'$ or for which the relevant path is longer than two elements, neither of which is the case for v and v^*, since $\pi^*(v)$ and $\pi^*(v^*)$ are neighbors. Hence, $s(v, v^*)$ and $s(v^*, v)$ remain in q' and we can choose the sequence $v_0, \ldots, v_n$ in Condition 3 as v, v^*, v and thereby satisfy the condition for the binary atom $r(v, v')$.

- The relevant path $d_0, \ldots, d_{n-1}$ from $\pi_1(v)$ to $\pi_1(v')$ is such that $n > 2$. By construction of q_1, it follows that there exists an $s \in \mathsf{Trans}$ such that

$s \sqsubseteq^* r$ and $s(v_0, v_1), \ldots, s(v_{n-2}, v_{n-1})$ in q_1, where $v_0, \ldots, v_{n-1} \in V$ are the variables such that $v_i = d_i$. Clearly, $v_i \neq v_j$ for all i, j with $i < j < n$ and the relevant path between d_i and d_{i+1} consists of only two nodes for all $i < n - 1$. Therefore $s(v_i, v_{i+1}) \in q'$ for all $i < n - 1$. It follows that Condition 3 is satisfied for the binary atom $r(v, v')$.

Since π and π' are clearly ε-compatible, it remains to prove that q' is tree-shaped. Let $v_r \in \mathsf{Var}(q')$ such that, for all $v \in \mathsf{Var}(q')$, $\pi'(v_r)$ is a (not necessarily proper) prefix of $\pi'(v)$. Such a variable exists due to Fact (a) above. Define a *trace* to be a sequence $t = v_0 \cdots v_n \in \mathsf{Var}(q')^+$ such that

- $v_0 = v_r$;
- for all $i < n$, $\pi'(v_i)$ is a true prefix of $\pi'(v_{i+1})$ and the length of the relevant path from $\pi'(v_i)$ to $\pi'(v_{i+1})$ is 2.

Let f be an injection from $\mathsf{Var}(q')$ to $\mathbb{N}$ and define $\tau(t) := \varepsilon \cdot f(v_1) \cdots f(v_n)$. It is easily seen that $T = \{\tau(t) \mid t \text{ is a trace}\}$ is a tree. For a trace $t = v_0 \cdots v_n$, let $\mathsf{last}(t) = v_n$. Clearly, for every variable $v \in \mathsf{Var}(q')$ there is a unique trace t_v such that $\mathsf{last}(t_v) = v$. Define a mapping $\nu : \mathsf{Var}(q') \to T$ by setting $\nu(v) := \tau(t_v)$. It is not difficult to verify that ν is a bijection. Let $r(v, v') \in q'$. By construction of q', this implies that the length of the relevant path from $\pi'(v)$ to $\pi'(v')$ is exactly 2. It is not hard to check that thus, $\nu(v)$ and $\nu(v')$ are neighbors in T. ❑

Lemma 11. *Let $\mathcal{I}$ be an interpretation, q a query, $q' \in \mathsf{tt}_{\mathcal{K}}(q)$, and π' a mapping such that $\mathcal{I} \models^{\pi'} q'$. Then there is a π such that $\mathcal{I} \models^{\pi} q$ and π and π' are ε-compatible.*

Proof. Let q_0 be a collapsing of q, q_1 an extension of q_0, and q' a tree-shaped reduction of q_1. Further suppose that $\mathcal{I} \models^{\pi'} q'$. We first show that $\mathcal{I} \models^{\pi'} q_1$. Let $at \in q_1$. If $at \in q'$, then $\mathcal{I} \models^{\pi'} q'$ implies $\mathcal{I} \models^{\pi'} at$. Otherwise, at is of the form $r(v, v')$ and there is an $s \in \mathsf{Trans}$ such that $s \sqsubseteq^* r$ and there are $v_0, \ldots, v_{n-1}$ such that $v_0 = v$, $v_{n-1} = v'$, and $s(v_i, v_{i+1}) \in q'$ for all $i < n - 1$. Since $\mathcal{I} \models^{\pi'} q'$, we have $(\pi(v), \pi(v_{i+1})) \in s^{\mathcal{I}}$ for all $i < n - 1$. Since $s \in \mathsf{Trans}$ and $s \sqsubseteq^* r$, we have $(\pi(v), \pi(v')) \in r^{\mathcal{I}}$ as required. Summing up, we have shown that $\mathcal{I} \models^{\pi'} q_1$. Since $q_0 \subseteq q_1$, this implies $\mathcal{I} \models^{\pi'} q_0$. Now, let π be obtained from π' by setting $\pi(v') = \pi'(v)$ if v is the variable that v' has been identified with when collapsing q to q_0. It is easy to check that $\mathcal{I} \models^{\pi} q$ and that π and π' are ε-compatible. ❑

Let q be a conjunctive query. It is easy to see how to produce the set S of all reducts of extensions of collapsings of q. To select the tree-shaped queries from S, we may proceed as follows. Let $q' \in S$ and select a $v_r \in \mathsf{Var}(q')$. Then define a mapping $\tau : \mathsf{Var}(q') \to \mathbb{N}^*$ inductively as follows:

- Initially, set $\tau(v_r) := \varepsilon$;

- if $\tau(v)$ is already defined and

$$V = \{v' \in \mathsf{Var}(q) \mid r(v, v') \text{ for some role } r \text{ and } \tau(v') \text{ undefined}\},$$

  then fix an injection $f : V \to \mathbb{N}$ and set $\tau(v') = \tau(v) \cdot f(v')$ for all $v' \in V$.

Clearly, $\mathsf{ran}(\tau)$ is a tree. The following is not difficult to prove.

Lemma 12. *The query q' is tree-shaped iff for all $r(v, v') \in q'$, $\tau(v)$ is a neighbor of $\tau(v')$.*

The algorithm to be designed in the following section crucially relies on the observation that tree-shaped queries can be converted into concepts formulated in the description logic $\mathcal{ELI}^\sqcap$, which offers only the concept constructors $\sqcap$ and $\exists r_0 \sqcap \cdots \sqcap r_{n-1}.C$, where $r_0, \dots, r_{n-1}$ are (possibly inverse or non-simple) roles. The semantics of the latter operator is as follows:

$$(\exists r_0 \sqcap \cdots \sqcap r_{n-1}.C)^\mathcal{I} := \{d \in \Delta^\mathcal{I} \mid \exists e : (d, e) \in r_i^\mathcal{I} \text{ for } 0 \leqslant i < n \text{ and } e \in C^\mathcal{I}\}.$$

More precisely, this conversion can be done as follows. Let q be a tree-shaped query and $\tau : \mathsf{Var}(q) \to \mathbb{N}^*$ with $\varepsilon \in \mathsf{ran}(\tau)$ such that $r(v, v') \in q$ iff $\tau(v)$ is a neighbor of $\tau(v')$. Then assign to each variable v a concept $C_q(v)$ by proceeding in a bottom-up fashion through the tree $\mathsf{ran}(\tau)$:

- if $\tau(v)$ is a leaf of $\mathsf{ran}(\tau)$, then $C_q(v) := \bigsqcap_{A(v) \in q} A$
- if $\tau(v)$ has successors $\tau(v_0), \dots, \tau(v_{n-1})$, then

$$C_q(v) := \bigsqcap_{A(v) \in q} A \sqcap \bigsqcap_{0 \leqslant i < n} \exists (\bigsqcap_{r(v, v_i) \in q} r).C_q(v_i).$$

Then C_q is $C_q(v_r)$ for $\tau(v_r) = \varepsilon$.

Lemma 13. *Let q be a tree-shaped query, $\mathcal{I}$ an interpretation, and $v_r \in \mathsf{Var}(q)$. Then $\mathcal{I} \models q$ iff $C_q(v_r)^\mathcal{I} \neq \emptyset$. In particular, $d \in C_q(v_r)^\mathcal{I}$ implies that there is a π with $v_r \mapsto d$ such that $\mathcal{I} \models^\pi q$.*

Lemma 13 shows that for all queries q, interpretations $\mathcal{I}$, and variables $v, v' \in \mathsf{Var}(q)$, we have $C_q(v)^\mathcal{I} \neq \emptyset$ iff $C_q(v')^\mathcal{I} \neq \emptyset$. This justifies the following: given a conjunctive query q, we use C_q to denote $C_q(v)$ for some arbitrary (but fixed) $v \in \mathsf{Var}(q)$.

We could now apply tree transformations to the sub-queries identified by a forest match, and use so-called representative concepts [8] or name formulae [3] to roll up the resulting query into a concept C_q. This would allow us to straightforwardly obtain a decision procedure: $\mathcal{K} \models q$ iff for every model $\mathcal{I}$ of $\mathcal{K}$ there is some C such that C is a concept that can be obtained by rolling-up a tree transformation of the sub-queries identified by a forest match of a transitive rewriting of q, and $C^{\mathcal{I}} \neq \emptyset$. If $\mathbf{C}$ is the set of all such concepts, then for $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$, $\mathcal{K} \models q$ iff $(\mathcal{T}', \mathcal{H}, \mathcal{A})$ is inconsistent, where

$$\mathcal{T}' = \mathcal{T} \cup \{\top \sqsubseteq \neg C \mid C \in \mathbf{C}\}.$$

By doing so, however, we would compromise the clear separation between the TBox and the ABox, and thus we could no longer obtain tight data complexity results. We will therefore present a decision procedure that uses extended ABoxes to check for the existence of forest matches; this decision procedure yields the desired complexity results.

4 The Decision Procedure

In order to gain insight into the data complexity of query entailment, we devise a procedure that uses extensions of both TBox and ABox. We proceed as follows: roughly speaking, we look for a KB $\mathcal{K}'$ such that $\mathcal{K}'$ extends $\mathcal{K}$ (both w.r.t. TBox and Abox), and the additional axioms and assertions prevent the existence of a transitive rewriting q' of q, a canonical model $\mathcal{I}$ of $\mathcal{K}$, and a forest match π such that $\mathcal{I} \models^{\pi} q'$. Lemmas 4 and 6 and the fact that $\mathcal{K}'$ extends $\mathcal{K}$ clearly also implies $\mathcal{K} \not\models q$. We consider all "relevant" extensions of $\mathcal{K}$ so that, if we find no extension $\mathcal{K}'$ such that $\mathcal{K}' \not\models q$, we can conclude that $\mathcal{K} \models q$.

In order to define $\mathcal{K}'$, we use Lemma 13, and thus $\mathcal{K}'$ will not be a $\mathcal{SHIQ}$ knowledge base. An *extended knowledge base* $\mathcal{K}'$ is of the form $(\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ with

- $\mathcal{T}$, $\mathcal{H}$, and $\mathcal{A}$ are as in a $\mathcal{SHIQ}$ knowledge base;
- $\mathcal{T}_q$ is a finite set of GCIs $\top \sqsubseteq C$ with C a $\mathcal{SHIQ}^{\sqcap}$ concept;
- $\mathcal{A}'$ is an ABox such that if $C(a) \in \mathcal{A}'$, then $a \in \mathsf{Ind}(\mathcal{A})$ and C is a negated $\mathcal{SHIQ}^{\sqcap}$ concept.

The extended knowlegde bases $\mathcal{K}'$ that we construct from $\mathcal{K}$ and q will be such that every counter model against $\mathcal{K} \models q$ (i.e., $\mathcal{I} \not\models q$) is a model of some $\mathcal{K}'$ and, for each model $\mathcal{I}$ of a $\mathcal{K}'$, we have that $\mathcal{I} \not\models q$. Thus, $\mathcal{K} \models q$ iff each $\mathcal{K}'$ is inconsistent. From Lemmas 4 and 6, to ensure that models of the $\mathcal{K}'$ are counter models, it suffices to prevent forest matches of transitive rewritings of q w.r.t. $\mathcal{K}$ in canonical models of $\mathcal{K}'$—and this is the role played by $\mathcal{T}_q$ and $\mathcal{A}'$. We distinguish between two kinds of forest matches: a-tree matches and *true* forest matches, i.e., forest matches that are not a-tree matches. To prevent a-tree matches, it suffices to consider the tree transformations of q. Therefore, $\mathcal{T}_q$

is defined as follows:

$$\mathcal{T}_q = \{\top \sqsubseteq \neg C_{q'} \mid q' \in \mathsf{tt}_{\mathcal{K}}(q)\}.$$

To prevent true forest matches, we further include an ABox $\mathcal{A}'$, which contains additional information about the individuals in $\mathcal{A}$. This is similar to the well-known precompletion approach for reducing ABox consistency to concept satisfiability [6]. Each $\mathcal{A}'$ represents a model in which there is no "true forest match" of a transitive rewriting of q, i.e., it contains, for each possible forest match, an assertion that "spoils" it.

This can consist either of an assertion which ensures that, for some grounded variable $v \in \mathsf{Var}(q')$ with $\pi(v) = (a, \varepsilon)$, a is not an instance of any rolling-up of a tree transformation of the a-tree match containing v, or of an assertion that ensures, for some grounded variables $v, v' \in \mathsf{Var}(q')$, with $r(v, v') \in q'$, $\pi(v) = (a, \varepsilon)$ and $\pi(v') = (b, \varepsilon)$, a is not r-related to b (i.e., $\neg r(a, b)$).

In the following, a *sub-query* of q is simply a non-empty subset of q (including q itself). Let Q be the set of all queries that are a tree transformation of a sub-query of a transitive rewriting of q w.r.t. $\mathcal{K}$, and let $\mathsf{cl}(q)$ be the set of all $C_{q'}$ such that $q' \in Q$. Note that this implies that $\mathsf{cl}(q)$ contains every concept name occurring in q.

An ABox $\mathcal{A}'$ is called a *q-completion* if it contains only assertions of the form

- $\neg C(a)$ for some $C \in \mathsf{cl}(q)$ and $a \in \mathsf{Ind}(\mathcal{A})$ and
- $\neg r(a, b)$ for a role name r occurring in $\mathsf{cl}(q)$ and $a, b \in \mathsf{Ind}(\mathcal{A})$.

Let $n = |q|, m_{\mathcal{H}} = |\mathcal{H}|, m_{\mathcal{A}} = |\mathcal{A}|$, and $k = |\mathsf{cl}(q)|$. By Lemmas 7 and 9 and since the number of sub-queries of q is bounded by 2^n, there is a polynomial p such that $k \leq 2^{p(n) \cdot \log p(m_{\mathcal{H}})}$. Also by Lemmas 7 and 9, there is a polynomial p' such that the size of each concept in $\mathsf{cl}(q)$ is bounded by $p'(n)$. Therefore, the number of q-completions is bounded by $2^{km_{\mathcal{A}} + 2km_{\mathcal{A}}^2}$.

Let q' be a transitive rewriting of q, and $\tau : \mathsf{Var}(q') \to \mathsf{Ind}(\mathcal{A})$ be a partial mapping. For $a \in \mathsf{Ind}(\mathcal{A})$, we set $\mathsf{Root}(a, \tau) = \{v \in \mathsf{Var}(q') \mid \tau(v) = a\}$, and we use $\mathsf{Reach}(a, \tau)$ to denote the set of variables $v \in \mathsf{Var}(q')$ for which there exists a sequence of variables $v_0, \ldots, v_{n-1}$, $n \geq 1$, such that

- $\tau(v_0) = a$ and $v_{n-1} = v$
- $\{v_0, \ldots, v_{n-1}\} \cap \mathsf{dom}(\tau) \subseteq \mathsf{Root}(a, \tau)$;
- for all $i < n - 1$, there is a role r s.t. $r(v_i, v_{i+1}) \in q$.

Observe that $\mathsf{Root}(a, \tau) = \mathsf{dom}(\tau) \cap \mathsf{Reach}(a, \tau)$.

We call τ a *split mapping* if $\mathsf{dom}(\tau) \neq \emptyset$ and, for all $a, b \in \mathsf{Ind}(\mathcal{A})$, $a \neq b$ implies $\mathsf{Reach}(a,\tau) \cap \mathsf{Reach}(b,\tau) = \emptyset$. Each split mapping τ induces, for each $a \in \mathsf{ran}(\tau)$, a sub-query q_a as follows:

$$\begin{aligned} q_a = \ & \{at \in q \mid \mathsf{Var}(\{at\}) \subseteq \mathsf{Reach}(a,\tau)\}\backslash \\ & \{r(v,v') \in q' \mid v, v' \in \mathsf{Root}(a,\tau)\}. \end{aligned}$$

An *extended* query is a query where disjunctions of $\mathcal{ELI}^{\sqcap}$ concepts can occur in concept atoms. From a transitive rewriting $q' \in \mathsf{tr}_{\mathcal{K}}(q)$ and a split mapping $\tau : \mathsf{Var}(q') \to \mathsf{Ind}(\mathcal{A})$ we obtain a *groundable rewriting* q'' of q' as follows:

- drop all atoms in q' which contain a variable $v \notin \mathsf{dom}(\tau)$;
- for each $a \in \mathsf{ran}(\tau)$, replace all variables $v \in \mathsf{Root}(a,\tau)$ with a new variable v_a; and
- for each $a \in \mathsf{ran}(\tau)$, let q_a be the sub-query of q' induced by τ, replace all $v \in \mathsf{Root}(a,\tau)$ with v_a and add $(C_{q_a^1} \sqcup \ldots \sqcup C_{q_a^m})(v_a)$, where each q_a^i is a tree transformation of q_a in which v_a was not replaced and $C_{q_a^i} = C_{q_a^i}(v_a)$.

In this case, we call τ the *grounding of* q'' and use $\tau(q'')$ for the result of replacing each v_a in q'' with a.

We say that a q-completion $\mathcal{A}'$ *spoils* $\tau(q'')$ if there is some

- $r(a,b) \in \tau(q'')$ and $\neg r(a,b) \in \mathcal{A}'$ or
- $(C_{q_a^1} \sqcup \ldots \sqcup C_{q_a^m})(a) \in \tau(q'')$ and $\neg C_{q_a^i}(a) \in \mathcal{A}'$ for $1 \leq i \leq m$.

Finally, a q-completion $\mathcal{A}'$ is called a *counter candidate* for q and $\mathcal{K}$ if, for all groundable rewritings q'' of transitive rewritings q' of q with grounding τ, $\mathcal{A}'$ spoils $\tau(q'')$.

Let us estimate the complexity of checking whether a given q-completion is a counter candidate. By Lemma 7, there is a polynomial p such that there are $2^{p(n)\cdot\log p(m_{\mathcal{H}})}$ transitive rewritings of q and it is easily seen that all tree transformations can be computed in this time bound as well. The number of q-completions (and therefore of counter candidates) is bounded by $2^{km_{\mathcal{A}}+2km_{\mathcal{A}}^2}$. Moreover, for $q' \in \mathsf{tr}(q)$, it can be decided in time polynomial in n and $m_{\mathcal{A}}$ whether a partial mapping τ is a split mapping for q' and $\mathcal{A}$, and there are at most $(m_{\mathcal{A}}+1)^{|q'|}$ such partial mappings. In order to check whether a q-completion is a counter candidate, we have to check for the existance of certain concepts C such that $C(a) \in \mathcal{A}'$. Clearly, the number of concepts relevant here is bounded by the cardinality of $\mathsf{cl}(q)$, which is bounded by $2^{p(n)\cdot\log p(m_{\mathcal{H}})}$ for a polynomial p. Together with Lemma 7, this implies that there is a polynomial p' such that checking whether a q-completion is a counter candidate can be done in time $2^{p(n)\cdot\log p(m_{\mathcal{H}})\cdot\log p(m_{\mathcal{A}})}$.

The following lemma forms the base of our decision procedure. A proof can be found in Section A.

Lemma 14. *$\mathcal{K} \not\models q$ iff there exists a counter candidate $\mathcal{A}'$ for q and $\mathcal{K}$ such that the extended knowledge base $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ is consistent.*

Intuitively, counter candidates are those q-completions that do not give rise to true forest matches. Since we prevent tree matches via the TBox $\mathcal{T}_q$, the knowledge bases $\mathcal{K}'$ of Lemma 14 capture exactly the counter models against $\mathcal{K} \models q$.

Based on this lemma, we define two versions of our decision procedure for query entailment in $\mathcal{SHIQ}$. The first version is deterministic and provides us with an upper bound for combined complexity, where all three components of the input knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ and the query are considered as the input. The second version is non-deterministic and yields a tight upper bound for data complexity, where $\mathcal{T}$, $\mathcal{H}$, and q are assumed fixed, and only $\mathcal{A}$ is the input. For the deterministic version, we make use of the following result which is proved in the appendix.

Theorem 15. *Given an extended knowledge base $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$, where $|(\mathcal{T}, \mathcal{H}, \mathcal{A})| = r$, the cardinality of $\mathcal{T}_q \cup \mathcal{A}'$ is s, and the maximum length of concepts in $\mathcal{T}_q$ and $\mathcal{A}'$ is t, we can decide consistency of $\mathcal{K}$ in deterministic time $2^{2^{p(t \cdot \log r \cdot \log s)}}$ with p a polynomial.*

The deterministic version of our algorithm is as follows: generate all q-completions of $\mathcal{A}$ and then check whether all extended knowledge bases that are induced by the counter candidates are inconsistent. By Lemma 14, this algorithm is correct. Observe that for all extended knowledge bases $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ whose inconsistency needs to be checked, the cardinality of $\mathcal{T}_q$ is bounded by k, the cardinality of $\mathcal{A}'$ is bounded by $km_{\mathcal{A}} + 2km^2_{\mathcal{A}}$, and hence the cardinality of $\mathcal{T}_q \cup \mathcal{A}'$ is bounded by $k + km_{\mathcal{A}} + 2km^2_{\mathcal{A}}$ (where $k = |\mathsf{cl}(q)|$), and (due to Parts (b) of Lemmas 7 and 9) the maximum length of concepts in $\mathcal{T}_q$ and $\mathcal{A}'$ is bounded by $p(n)$ for some polynomial p. This together with Theorem 15, the bound established above on the number of q-completions of $\mathcal{A}$, and the fact that deciding if a q-completion is a counter candidate can be checked in time $2^{p(n) \cdot \log p(m_{\mathcal{H}}) \cdot \log p(m_{\mathcal{A}})}$ with p a polynomial, yields the following result.

Theorem 16. *Given a $\mathcal{SHIQ}$ knowledge base $\mathcal{K}$ and a conjunctive query q with $|\mathcal{K}| = m$ and $|q| = n$, it can be decided in deterministic time $2^{2^{p(n) \cdot \log p(m)}}$ whether $\mathcal{K} \models q$, where p is a polynomial.*

Observe that this bound is single exponential in the size of the knowledge base and double exponential in the size of the query.

The non-deterministic version of our decision procedure actually decides *non*-entailment of queries: guess a q-completion of $\mathcal{A}$, check whether it is a counter candidate and consistent, return "yes" ($\mathcal{K} \not\models q$) if the check succeeds and "no" ($\mathcal{K} \models q$) otherwise. Regarding the complexity of inconsistency, we make use of the following result which is also proved in the appendix.

Theorem 17. *Let $\mathcal{T}$ and $\mathcal{T}_q$ be TBoxes and $\mathcal{H}$ a role hierarchy. Given ABoxes $\mathcal{A}$ and $\mathcal{A}'$ such that $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ is an extended knowledge base and $|\mathcal{A} \cup \mathcal{A}'| = r$, we can decide in non-deterministic time $p(r)$ whether $\mathcal{K}'$ is consistent.*

Again, Lemma 14 yields correctness of our algorithm. Let $m_{\mathcal{A}} = |\mathcal{A}|$. The bound established above on the maximal size of q-completions implies that q-completions of $\mathcal{A}$ are polynomial in $m_{\mathcal{A}}$. Whether a q-completion is a counter candidate can be decided in time $2^{p(n)\cdot \log p(m_{\mathcal{H}})\cdot \log p(m_{\mathcal{A}})}$, which is also polynomial in $m_{\mathcal{A}}$. Thus, Theorem 17 implies that the data complexity of query entailment in $\mathcal{SHIQ}$ is in co-NP. The lower bound easily follows from the fact that conjunctive query entailment is already co-NP-hard regarding data complexity in the very restricted DL $\mathcal{AL}$ [2].

Theorem 18. *Conjunctive query entailment in $\mathcal{SHIQ}$ is data complete for co-*NP.

A Correctness

Our aim is to prove Lemma 14. To this end, we first prove that every extended knowledge base that is consistent has a canonical model.

Lemma 19. *Every extended knowledge base that is consistent has a canonical model.*

Proof. We translate an extended knowledge base $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ into an equisatisfiable $\mathcal{ALCQI}b$ knowledge base by extending the translation given in [12, 6.22]. Checking the consistency of an $\mathcal{ALCQI}b$ knowledge base is reduced (via precompletions) to $\mathcal{ALCQI}b$ concept satisfiability. Since the satisfiability of an $\mathcal{ALCQI}b$ concept is reduced to an emptiness test of an infinite automaton, it immediately follows that every $\mathcal{ALCQI}b$ concept has a canonical model. This yields that also every consistent extended knowledge base has a canonical model. ❑

We can now prove Lemma 14.

Lemma 14. $\mathcal{K} \not\models q$ iff there is some counter candidate $\mathcal{A}'$ of $\mathcal{A}$ such that the extended knowledge base $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ is consistent.

Proof. "⇒". Assume that $\mathcal{K} \not\models q$. We now have to show that there is a counter candidate $\mathcal{A}'$ of $\mathcal{A}$ such that the extended knowledge base $\mathcal{K} = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ is consistent. Since $\mathcal{K} \not\models q$, there is a model $\mathcal{I}$ of $\mathcal{K}$ such that $\mathcal{I} \not\models q$. We first show that $\mathcal{I}$ is a model of $\mathcal{T}_q$. To this end, let $\top \sqsubseteq \neg C$ in $\mathcal{T}_q$. Then $C = C_{q'}$ for some $q' \in \mathsf{tt}_{\mathcal{K}}(q)$. Assume to the contrary of what is to be shown that $C_{q'}^{\mathcal{I}} \neq \emptyset$. Then we get $\mathcal{I} \models q'$ by Lemma 13 and $\mathcal{I} \models q$ by Lemma 11, which is a contradiction.

Next, we define a q-completion $\mathcal{A}'$ such that

- if $C \in \mathsf{cl}(q)$, $a \in \mathsf{Ind}(\mathcal{A})$, and $a^{\mathcal{I}} \in \neg C^{\mathcal{I}}$, then $\neg C(a) \in \mathcal{A}'$;
- if the role name r occurs in $\mathsf{cl}(q)$, $a, b \in \mathsf{Ind}(\mathcal{A})$, and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin r^{\mathcal{I}}$, then $\neg r(a, b) \in \mathcal{A}'$.

Clearly, $\mathcal{I} \models \mathcal{A}'$ and therefore $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ is consistent. Thus, it remains to be shown that $\mathcal{A}'$ is a counter candidate of $\mathcal{A}$. Assume to the contrary that there is a transitive rewriting $q' \in \mathsf{tr}_{\mathcal{K}}(q)$, a split mapping τ for q' and $\mathcal{A}'$, and $\mathcal{A}'$ does not spoil $\tau(q'')$ for a groundable rewriting q'' of q'. For each $a \in \mathsf{ran}(\tau)$, let q_a be the sub-query of q' induced by τ and $\mathcal{A}$. Since $\mathcal{A}'$ does not spoil $\tau(q'')$, we have that for all $a \in \mathsf{ran}(\tau)$,

(i) q_a is empty or

(ii) there is a tree transformation q_a^i of q_a w.r.t. $\mathcal{K}$ such that $\neg C(a) \notin \mathcal{A}'$ where $C = C_{q_a^i}(v_a)$.

For all a such that (ii) holds, we therefore have, together with the definition of $\mathcal{A}'$ and the fact that each $C_{q_a^i}(v_a)$ is in $\mathsf{cl}(q)$, that $a^{\mathcal{I}} \in C$ with $C = C_{q_a^i}(v_a)$ for some tree transformation q_a^i of q_a.

Let $a_1, \ldots, a_n$ be the elements of $\mathsf{ran}(\tau)$ for which (ii) is true. Since $\mathcal{I} \models \mathcal{A}'$ and by Lemmas 13 and 11, there are mappings $\pi_1, \ldots, \pi_n$ such that for $1 \leq i \leq n$, we have

(a) $\mathcal{I} \models^{\pi_i} q_{a_i}$;

(b) $\pi_i(v_{a_i}) = a_i^{\mathcal{I}}$.

We now define a mapping $\pi : \mathsf{Var}(q') \to \Delta^{\mathcal{I}}$ as follows:

$$\pi(v) := \begin{cases} a^{\mathcal{I}} & \text{if } v \in \mathsf{Root}(a, \tau) \\ \pi_i(v) & \text{if } v \in \mathsf{Reach}(a_i, \tau) \setminus \mathsf{Root}(a_i, \tau) \end{cases}$$

Using the definition of $\mathsf{Reach}(a_i,\tau)$ and $\mathsf{Root}(a_i,\tau)$, it can be seen that π is well-defined, i.e., that the different cases are mutually exclusive: first, if $\tau(v)=a$, then $v\in\mathsf{Reach}(a,\tau)$ implies $a=a_i$ and $v=v_{a_i}$, and thus $v\notin\mathsf{Reach}(a_i,\tau)\setminus\mathsf{Root}(a_i,\tau)$; and second, the definition of the sets Reach ensures that $i\neq j$ implies ($a_i\neq a_j$ and thus) $\mathsf{Reach}(a_i,\tau)\cap\mathsf{Reach}(a_j,\tau)=\emptyset$.

Also observe that π is total on $\mathsf{Var}(q')$ because each $v\in\mathsf{Var}(q')$ is either in the range of τ or in one of the sets $\mathsf{Reach}(a_1,\tau),\dots,\mathsf{Reach}(a_n,\tau)$.

To finish the proof, we show that $\mathcal{I}\models^\pi q'$. This yields $\mathcal{I}\models q$ by Lemma 6 and thus the desired contradiction. First, let $A(v)\in q'$. There are two cases:

- $v\in\mathsf{dom}(\tau)$. Let $a_i=\tau(v)$, then by definition of $\mathsf{Reach}(a_i,\tau)$, $v\in\mathsf{Reach}(a_i,\tau)$, and hence $A(v)\in q_{a_i}$. Then (a), (b), and the definition of π yield $\mathcal{I}\models^\pi A(v)$ as required.
- $v\notin\mathsf{dom}(\tau)$. Then connectedness of q' and the definition of the sets $\mathsf{Reach}(a_i,\tau)$ implies that $v\in\mathsf{Reach}(a_i,\tau)\setminus\mathsf{Root}(a_i,\tau)$ for some i with $1\leq i\leq n$. Then we have $A(v)\in q_{a_i}$ and (a) together with the definition of π yields $\mathcal{I}\models^\pi A(v)$.

Next, let $r(v,v')\in q'$. There are four cases:

- $v,v'\in\mathsf{dom}(\tau)$. By construction of q'' from q' and since $\mathcal{A}'$ does not spoil $\tau(q'')$ by assumption, we have that $r(\tau(v),\tau(v'))\in\mathcal{A}'$. Since $\mathcal{I}\models\mathcal{A}'$ and by definition of π, we have that $\mathcal{I}\models^\pi r(v,v')$.
- $v\in\mathsf{dom}(\tau)$ and $v'\notin\mathsf{dom}(\tau)$. By definition of the sets $\mathsf{Reach}(a_i,\tau)$, this means that for some i with $1\leq i\leq n$, we have $\tau(v)=a_i$ and $v'\in\mathsf{Reach}(a_i,\tau)\setminus\mathsf{Root}(a_i,\tau)$. Due to the definition of q_{a_i}, it follows that $r(v,v')\in q_{a_i}$. Now (a), (b), and the definition of π yield $\mathcal{I}\models^\pi r(v,v')$ as required.
- $v\notin\mathsf{dom}(\tau)$ and $v'\in\mathsf{dom}(\tau)$. Symmetric to the previous case.
- $v,v'\notin\mathsf{dom}(\tau)$. Then connectedness of q' and the definition of the sets $\mathsf{Reach}(a_i,\tau)$ implies that $v,v'\in\mathsf{Reach}(a_i,\tau)\setminus\mathsf{Root}(a_i,\tau)$ for some i with $1\leq i\leq n$. We can continue as in the second case for atoms $A(v)$.

"$\Leftarrow$". Assume that there is a counter candidate $\mathcal{A}'$ of $\mathcal{A}$ such that $\mathcal{K}'=(\mathcal{T}\cup\mathcal{T}_q,\mathcal{H},\mathcal{A}\cup\mathcal{A}')$ is consistent. We now have to show that $\mathcal{K}\not\models q$. By Lemma 19, there is a model $\mathcal{I}=(\Delta^\mathcal{I},\cdot^\mathcal{I})$ of $\mathcal{K}'$ that is a canonical intepretation. Since $\mathcal{I}$ is clearly also 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 6, there exists a transitive rewriting $q'\in\mathsf{tr}_\mathcal{K}(q)$ and a forest match π such that $\mathcal{I}\models^\pi q'$. We distinguish two cases.

First, assume that π is an a-tree match, i.e., there is an $a \in \mathsf{N_I}$ such that for all $v \in \mathsf{Var}(q')$, we have $\pi(v) = (a, w)$ for some $w \in \mathbb{N}^*$. Let $\mathcal{J}$ be the forest base of $\mathcal{I}$, $\mathcal{J}_a$ the a-tree base obtained from $\mathcal{J}$ by restricting the domain to tuples (a, w') with $w' \in \mathbb{N}^*$, and $\mathcal{I}_a$ the canonical interpretation based on $\mathcal{J}_a$. Since the range of π is a subset of $\Delta^{\mathcal{I}_a}$, it is not hard to see that $\mathcal{I} \models^\pi q'$ implies $\mathcal{I}_a \models^\pi q'$. By Lemma 6, $\mathcal{I}_a \models q$ and by Lemma 10, there is a $q^* \in \mathsf{tt}_\mathcal{K}(q)$ such that $\mathcal{I}_a \models q^*$. It is easy to see that this implies $\mathcal{I} \models q^*$. Finally, Lemma 13 yields that $C^{\mathcal{I}}_{q^*} \neq \emptyset$, which is a contradiction to $\mathcal{I}$ satisfying the concept inclusion $\top \sqsubseteq \neg C_{q^*}$.

Now assume that π is a true forest match. Recall that in order to show a contradiction, we assumed that $\mathcal{I} \models^\pi q'$ for some $q' \in \mathsf{tr}_\mathcal{K}(q)$. We now show that $\mathcal{A}'$ cannot be a counter candidate, i.e., there is a split mapping τ and groundable rewriting q'' of q' such that $\mathcal{A}'$ does not spoil $\tau(q'')$.

We define a mapping τ by setting

$$\tau(v) = a \text{ if } \pi(v) = (a, \varepsilon), \text{ for all } v \in \mathsf{Var}(q').$$

By definition of τ and since π is a forest match, τ is a split mapping. Now, let q'' be the groundable rewriting for q' and τ. We have to show that $\mathcal{A}'$ does not spoil $\tau(q'')$.

- Let $r(a, b) \in \tau(q'')$ and assume for a contradiction that $\neg r(a, b) \in \mathcal{A}'$. Since $r(a, b) \in \tau(q'')$, there are variables $v, v' \in \mathsf{dom}(\tau)$ such that $\tau(v) = a$ and $\tau(v') = b$ for $a, b \in \mathsf{Ind}(\mathcal{A})$ and $r(v, v') \in q'$. By definition of τ from π, $\pi(v) = (a, \varepsilon)$ and $\pi(v') = (b, \varepsilon)$. Since $\mathcal{I} \models^\pi q'$ and $r(v, v') \in q'$, $((a, \varepsilon), (b, \varepsilon)) \in r^\mathcal{I}$ and hence, by definition of $\mathcal{A}'$, $\neg r(a, b) \notin \mathcal{A}'$, which clearly is a contradiction.

- Let $(C_{q_a^1} \sqcup \ldots \sqcup C_{q_a^m})(a) \in \tau(q'')$ and assume for a contradiction that $\neg C_{q_a^i}(a) \in \mathcal{A}'$ for $1 \leq i \leq m$. Let q_a be the subquery induced by τ and a, and q_a' the result of replacing each $v \in \mathsf{Root}(a, \tau)$ with v_a. Since $(C_{q_a^1} \sqcup \ldots \sqcup C_{q_a^m})(a) \in \tau(q'')$, $(C_{q_a^1} \sqcup \ldots \sqcup C_{q_a^m})(v_a) \in q''$, q_a' is non-empty and $q_a^1, \ldots, q_a^m$ are all the tree transformations of q_a' in which v_a has not been replaced. Note also that $\pi(v_a) = (a, \varepsilon)$.

  Due to the fact that $\mathcal{I} \models^\pi q'$, it is easily seen that $\mathcal{I} \models^\pi q_a$. Since π is a forest match and since q_a contains only variables from $\mathsf{Reach}(a, \tau)$, the claim yields that π is even a tree match for q_a. Let $\mathcal{J}$ be the forest base of $\mathcal{I}$, $\mathcal{J}_a$ the a-tree base obtained from $\mathcal{J}$ by restricting the domain of $\mathcal{J}$ to pairs $(a, \cdot)$, and $\mathcal{I}_a$ the canonical interpretation based on $\mathcal{J}_a$. Since π maps all variables in q_a to $\Delta^{\mathcal{I}_a}$, it is not hard to see that $\mathcal{I} \models^\pi q_a$ implies $\mathcal{I}_a \models^\pi q_a$. By Lemma 10, there is a $q_a^i \in \mathsf{tt}_\mathcal{K}(q_a)$ such that $\mathcal{I}_a \models^{\pi'} q_a^i$ for some π' that is ε-compatible with π. It is easy to see that we also have $\mathcal{I} \models^{\pi'} q_a^i$. This together with the facts that $\pi(v_a) = (a, \varepsilon)$ and π and π'

are ε-compatible yields $a^{\mathcal{I}} = (a, \varepsilon) \in C^{\mathcal{I}}$ for $C = C_{q_a^i}(v_a)$ and hence, by construction of $\mathcal{A}'$, $\neg C(a) \notin \mathcal{A}'$, which clearly is a contradiction.

Since this argument was independent of the chosen $C(a) \in \tau(q'')$, we conclude that $\mathcal{A}'$ does not spoil $\tau(q'')$ as intended.

❏

B Complexity

Tobies [12, Corollary 6.30] showed that deciding knowlege base consistency for $\mathcal{SHIQ}$ is ExpTime-complete (even for binary coding of numbers) by reducing $\mathcal{SHIQ}$ knowledge base consistency to concept satisfiability testing in $\mathcal{ALCQI}b$. The b stands for safe Boolean role expressions build from $\mathcal{ALCQI}b$ roles using the operator $\sqcap$ (role intersection), $\sqcup$ (role union), and $\neg$ (role negation/complement) such that, when transformed into disjunctive normal form, every disjunct contains at least one non-negated conjunct. Given a query q and a $\mathcal{SHIQ}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$, we reduce query entailment to deciding knowledge base consistentcy of an extended $\mathcal{SHIQ}^{\sqcap}$ knowledge base $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$. Recall that $\mathcal{T}_q$ and $\mathcal{A}'$ are the only parts that contain role conjunctions.

We assume here, that all concepts are in *negation normal form (NNF)*; any concept can be transformed in linear time into an equivalent one in NNF by pushing negation inwards, making use of de Morgan's laws and the duality between existential and universal restrictions, and between atmost and atleast number restrictions ($\leqslant nr.C$ and $\geqslant nr.C$ respectively). For a concept C, we use $\dot{\neg} C$ to denote the NNF of $\neg C$.

We first recall some of the facts that we have shown before. Let $m = |\mathcal{K}|, m_{\mathcal{H}} = |\mathcal{H}|, m_{\mathcal{A}} = |\mathcal{A}|$, and $n = |q|$. By Lemmas 7 and 9, the number of transitive rewritings and of tree transformations for a query q is bounded by $2^{p(n) \cdot \log p(m_{\mathcal{H}})}$ for some polynomial q, the size of each transitive rewriting and of each tree transformation is polynomial in n, and hence the length of each concept in $\mathcal{T}_q$ and $\mathcal{A}'$ is also polynomial in n. Therefore, there is a polynomial p such that $|\mathcal{T}_q| \leqslant 2^{p(n) \cdot \log p(m_{\mathcal{H}})}$. Let $k = |\mathsf{cl}(q)|$. As we have shown before, $k \leqslant 2^{p(n) \cdot \log p(m_{\mathcal{H}})}$ for a polynomial p. A q-completion $\mathcal{A}'$ contains at most $km_{\mathcal{A}} + 2km_{\mathcal{A}}^2$ assertions each of size at most $p(n)$. Therefore, $2^{p'(n) \cdot \log p'(m_{\mathcal{H}}) \cdot \log p'(m_{\mathcal{A}})}$ is an upper bound for $|\mathcal{A}'|$ with p' a polynomial and it is easy to see that $|\mathcal{T}_q \cup \mathcal{A}'|$ is exponential in n and polynomilal in $m_{\mathcal{H}}$ and $m_{\mathcal{A}}$.

Let now $|\mathcal{T}_q \cup \mathcal{A}'| = s$ and t the maximal length of concepts in $\mathcal{T}_q \cup \mathcal{A}'$. Due to the above computations, we have $t \leqslant p(n)$ and $s \leqslant 2^{p(t) \cdot \log p(m_{\mathcal{H}}) \cdot \log p(m_{\mathcal{A}})}$ for p a polynomial.

Our aim is to translate $\mathcal{SHIQ}^{\sqcap}$ to $\mathcal{ALCQI}b$ and reuse the complexity results obtained by Tobies. Therefore, we first show how we can extend the translation given by Tobies, which encodes the role hierarchy by means of role conjunctions.

We first consider $\mathcal{SHIQ}^{\sqcap}$-concept satisfiability w.r.t. a role hierarchy and, by using “internalization”, this is enough in order to decide the satisfiability of a TBox with respect to a role hierarchy.

Definition 20. For $r, r_0, \dots, r_{n-1}$ roles, let

$$r^{\uparrow} = \prod_{r \sqsubseteq^* s} s$$

and

$$(r_0 \sqcap \dots \sqcap r_{n-1})^{\uparrow} = r_0^{\uparrow} \sqcap \dots \sqcap r_{n-1}^{\uparrow}.$$

△

Note that, since $r \sqsubseteq^* r$, r occurs in $r^{\uparrow}$.

Lemma 21. *Let $\mathcal{H}$ be a role hierarchy, and $\mathcal{R} = r_0 \sqcap \dots \sqcap r_{n-1}$ a conjunction of roles. For every interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{H}$, $\left((r_0 \sqcap \dots \sqcap r_{n-1})^{\uparrow}\right)^{\mathcal{I}} = (r_0 \sqcap \dots \sqcap r_{n-1})^{\mathcal{I}}$.*

Proof. By definition of $r^{\uparrow}$, $(r_0 \sqcap \dots \sqcap r_{n-1})^{\uparrow} = r_0^{\uparrow} \sqcap \dots \sqcap r_{n-1}^{\uparrow}$. By definition of the semantics of role conjunctions, we have that $\left(r_0^{\uparrow} \sqcap \dots \sqcap r_{n-1}^{\uparrow}\right)^{\mathcal{I}} = \left(r_0^{\uparrow}\right)^{\mathcal{I}} \cap \dots \cap \left(r_{n-1}^{\uparrow}\right)^{\mathcal{I}}$. If $s \sqsubseteq^* r$, then $\{s' \mid r \sqsubseteq^* s'\} \subseteq \{s' \mid s \subseteq^* s'\}$ and hence $\left(s^{\uparrow}\right)^{\mathcal{I}} \subseteq \left(r^{\uparrow}\right)^{\mathcal{I}}$. If $\mathcal{I} \models \mathcal{H}$, then $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ for every s with $r \sqsubseteq^* s$. Hence, $\left(r^{\uparrow}\right)^{\mathcal{I}} = r^{\mathcal{I}}$ and $\left(r_0^{\uparrow} \sqcap \dots \sqcap r_{n-1}^{\uparrow}\right)^{\mathcal{I}} = {r_0}^{\mathcal{I}} \cap \dots \cap {r_{n-1}}^{\mathcal{I}}$ as required. ❏

With the extended definition of ↑ on role conjunctions, we can now adapt the definition (Def. 6.22) that Tobies provides for translating $\mathcal{SHIQ}$-concepts into $\mathcal{ALCQI}b$-concepts.

Definition 22. Let C be a $\mathcal{SHIQ}^{\sqcap}$-concept and $\mathcal{H}$ a role hierarchy. For every concept $\forall(r_0 \sqcap \dots \sqcap r_{n-1}).D \in \mathsf{cl}(C)$, let $X_{r_0 \sqcap \dots \sqcap r_{n-1}, D} \in \mathsf{N_C}$ be a unique concept name that does not occur in $\mathsf{cl}(C)$. We define the function $\cdot^{tr}$ inductively on the structure of concepts by setting

$$\begin{array}{rcl}
A^{tr} & = & A \text{ for all A } \in \mathsf{N_C} \\
(\neg A)^{tr} & = & \neg A \text{ for all A } \in \mathsf{N_C} \\
(C_1 \sqcap C_2)^{tr} & = & C_1^{tr} \sqcap C_2^{tr} \\
(C_1 \sqcup C_2)^{tr} & = & C_1^{tr} \sqcup C_2^{tr} \\
(\bowtie n(r_0 \sqcap \dots \sqcap r_{n-1}).D)^{tr} & = & (\bowtie n(r_0 \sqcap \dots \sqcap r_{n-1})^{\uparrow}.D^{tr}) \\
(\forall(r_0 \sqcap \dots \sqcap r_{n-1}).D)^{tr} & = & X_{r_0 \sqcap \dots \sqcap r_{n-1}, D} \\
(\exists(r_0 \sqcap \dots \sqcap r_{n-1}).D)^{tr} & = & \neg X_{r_0 \sqcap \dots \sqcap r_{n-1}, \dot{\neg} D}
\end{array}$$

where $\bowtie$ stands for $\leqslant$ or $\geqslant$. We now define an extended TBox $\mathcal{T}_C$ by setting

$$\begin{array}{rcl}
\mathcal{T}_C & = & \{ X_{r_0 \sqcap \dots \sqcap r_{n-1}, D} \equiv \\
& & \quad \forall(r_0 \sqcap \dots \sqcap r_{n-1})^{\uparrow}.D^{tr} \mid \forall(r_0 \sqcap \dots \sqcap r_{n-1}).D \in \mathsf{cl}(\mathcal{K}) \} \cup \\
& & \{ X_{r_0 \sqcap \dots \sqcap r_{n-1}, D} \sqsubseteq \displaystyle\prod_{T \in \{t_0 \sqcap \dots \sqcap t_{n-1} \mid t_i \sqsubseteq^* r_i, t_i \in \mathsf{Trans}, i \leq n\}} \forall T^{\uparrow}.X_{T,D} \}
\end{array}$$

△

Lemma 23. *Let $\mathcal{H}$ be a role hierarchy, $C = C_{\mathcal{T}} \sqcap \dot{\neg} C_q$ with $C_{\mathcal{T}}$ a $\mathcal{SHIQ}$-concept and C_q an $\mathcal{ELI}^{\sqcap}$-concept, and $\cdot^{tr}$ and $C_{\mathcal{T}}$ defined as in Def. 22, then C is satisfiable w.r.t. $\mathcal{H}$ iff the $\mathcal{ALCQI}b$-concept C^{tr} is satisfiable w.r.t. $C_{\mathcal{T}}$.*

Proof. Given Lemma 21, the proof is a long, but straightforward extension of the proof given by Tobies [12, Lemma 6.23]. ❑

Observe, however, that the translation of a $\mathcal{SHIQ}$-concept C w.r.t. a role hierarchy $\mathcal{H}$ is polynomial in $|C|$ and $|\mathcal{H}|$, while for a $\mathcal{SHIQ}^{\sqcap}$-concept C it is polynomial in $|C|$, but exponential in $|\mathcal{H}|$, due to the last step in the definition of $C_{\mathcal{T}}$, which is required for handling all possible combinations of transitive subroles in the different conjuncts. Therefore, we would not obtain the ExpTime upper bound we are aiming for. However, for deciding query entailment, the input concept consists of the two parts $C_{\mathcal{T}}$ and C_q, where $C_{\mathcal{T}}$ is the internalization of the $\mathcal{SHIQ}$ TBox and C_q the internalization of the $\mathcal{SHIQ}^{\sqcap}$ TBox, i.e., C_q is a conjunction of negated concepts from $\mathsf{cl}(q)$ such that $|C_q| \leqslant s$. Only C_q contains role conjunctions, the number of roles in each conjunction is bounded by t, and $m_{\mathcal{H}}$ gives a bound on the number of roles in $\mathcal{H}$. Hence the size of C_q^{tr} is bounded by $2^{p(t) \cdot \log p(s) \cdot \log p(m_{\mathcal{H}})}$ for p a polynomial.

Theorem 15. Given an extended knowledge base $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ where $|(\mathcal{T}, \mathcal{H}, \mathcal{A})| = m$, the cardinality of $\mathcal{T}_q \cup \mathcal{A}'$ is s, and the maximum length of concepts in $\mathcal{T}_q$ and $\mathcal{A}'$ is t, we can decide consistency of $\mathcal{K}'$ in deterministic time $2^{2^{p(t \cdot \log r \cdot \log s)}}$ with p a polynomial.

Proof. Since the given translation works also for concepts in the ABox and since $\mathcal{ALCQI}b$ provides for negated roles, we can directly translate an extended $\mathcal{SHIQ}$ knowledge bases into an $\mathcal{ALCQI}b$ knowledge base with the translation given in Def. 22. For the $\mathcal{SHIQ}$ parts, the translation is again polynomial in the size of the knowledge base, i.e., for the $\mathcal{SHIQ}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ the size of the obtained $\mathcal{ALCQI}b$ knowledge base $\mathcal{K}^{tr} = (\mathcal{T}^{tr}, \mathcal{A}^{tr})$ is polynomial in m. However, the size of the translation for the $\mathcal{SHIQ}^{\sqcap}$ ABox $\mathcal{A}'$ is polynomial only in $m_{\mathcal{H}}$ and s, but exponential in t. For p a polynomial, we therefore obtain an upper bound of $2^{p(t) \cdot \log p(s) \cdot \log p(m)}$ for the size of the $\mathcal{ALCQI}b$ knowledge base $\mathcal{K}^{tr} = (\mathcal{T}^{tr} \cup \mathcal{T}_q^{tr}, \mathcal{A}^{tr} \cup \mathcal{A}'^{tr})$ obtained by applying the translation from Definition 22. Since deciding whether an $\mathcal{ALCQI}b$ knowledge base is consistent is an ExpTime-complete problem (even with binary coding of numbers) [12, Theorem 4.42], it can be checked in time $2^{2^{p(t) \cdot \log p(s) \cdot \log p(m)}}$ if $\mathcal{K}$ is consistent or not. ❑

Let $\mathcal{K} = (\mathcal{T},\ \mathcal{H},\ \mathcal{A})$ be a $\mathcal{SHIQ}$ knowledge base. In order to obtain an upper bound on the data complexity of the query entailment problem, we assume w.l.o.g. that all concept assertions in the ABox $\mathcal{A}$ are of a fixed size, i.e., for each $C(a) \in \mathcal{A}$, $|C| \leq t$ for some fixed t. We now give the proofs for Theorem 17 and Theorem 18.

Theorem 17. Let $\mathcal{K} = (\mathcal{T}, \mathcal{H}, \mathcal{A})$ be a $\mathcal{SHIQ}$ knowledge base, q a Boolean conjunctive query, $\mathcal{K}' = (\mathcal{T} \cup \mathcal{T}_q, \mathcal{H}, \mathcal{A} \cup \mathcal{A}')$ the extended knowledge base for $\mathcal{K}$ and q, and $r = |\mathcal{A} \cup \mathcal{A}'|$. Deciding whether $\mathcal{K}'$ is consistent can be done in non-deterministic time $p(r)$ for some polynomial p.

Proof. We assume $q, \mathcal{T}, \mathcal{T}_q$, and $\mathcal{H}$ to be fixed. Let $c = |\mathcal{T} \cup \mathcal{T}_q \cup \mathcal{H}|$ and $n = |q|$. By Lemmas 7 and 9 and since only the role hierarchy $\mathcal{H}$ increases the number of atoms in transitive rewritings and tree transformations (the ABox has no effect), the assumption that the size of $\mathcal{T}_q$ is fixed is valid.

Since the number of matching candidates is polynomial in r, checking if a q-completion is a counter-candidate can be done in time polynomial in r.

Since the translation of an extended knowledge base into an $\mathcal{ALCQIb}$ knowledge base is exponential only in n and since we assume n to be fixed, we can translate $\mathcal{K}'$ into an equisatisfiable $\mathcal{ALCQIb}$ knowledge base $\mathcal{K}'^{tr}$ in time polynomial in r.

In order to determine the consistency of an $\mathcal{ALCQIb}$ knowledge base, Tobies uses precompletions and in Lemma 4.40 Tobies shows that an $\mathcal{ALCQIb}$ knowledge base is consistent iff it has a precompletion as defined in Definition 4.39. Hence, we can guess such a precompletion $\mathcal{A}^*$ of $\mathcal{A}^{tr} \cup \mathcal{A}'^{tr}$ and a mapping f from individuals in $\mathcal{A}^*$ to $\mathcal{A}^{tr} \cup \mathcal{A}'^{tr}$ and since checking the conditions on precompletions can be done in time polynomial in r, it is then an immediate consequence of [12, Theorem 4.42] that checking consistency of $\mathcal{K}'$ can be done in non-deterministic time $p(r)$ for some polynomial p. ❑

Theorem 18. Conjunctive query entailment in $\mathcal{SHIQ}$ is data complete for co-NP.

Proof. The lower bound immediately follows from the fact that conjunctive query answering is already co-NP-hard regarding data complexity in the very restricted DL $\mathcal{AL}$ [2]. The upper bound is a consequence of Lemma 14 and Theorem 17. ❑

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementa-*

tion, and Applications. Cambridge University Press, 2003.

[2] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proceedings of the 18th International Workshop on Description Logics (DL 2005)*, 2005.

[3] D. Calvanese, G. D. Giacomo, and M. Lenzerini. On the decidability of query containment under constraints. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)*, pages 149–158. ACM Press, 1998.

[4] B. Glimm, I. Horrocks, and U. Sattler. Conjunctive query answering for description logics with transitive roles. In *Proc. of DL 06*. CEUR, 2006.

[5] V. Haarslev, R. Möller, R. V. D. Straeten, and M. Wessel. Extended query facilities for racer and an application to software-engineering problems. In *Proc. of DL 04*. CEUR, 2004.

[6] B. Hollunder. Consistency checking reduced to satisability of concepts in terminological systems. *Annals of Mathematics and Artificial Intelligence*, 18(2–4):133–157, 1996.

[7] I. Horrocks, P. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1), 2003.

[8] I. Horrocks and S. Tessaris. A conjunctive query language for description logic aboxes. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI 2000)*, pages 399–404, 2000.

[9] U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In *Proceedings of the Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, pages 466–471, 2005.

[10] B. Motik, U. Sattler, and R. Studer. Query answering for OWL-DL with rules. In *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, Hiroshima, Japan, November 2004.

[11] M. M. Ortiz, D. Calvanese, and T. Eiter. Characterizing data complexity for conjunctive query answering in expressive description logics. In *Proc. of AAAI 2006*, 2006. To appear.

[12] S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation.* PhD thesis, RWTH Aachen, 2001.