Answering Conjunctive Queries over $\mathcal{EL}$ Knowledge Bases with Transitive and Reflexive Roles

Giorgio Stefanoni and **Boris Motik**
Department of Computer Science, University of Oxford
Wolfson Building, Parks Road,
Oxford, OX1 3QD, UK

Abstract

Answering conjunctive queries (CQs) over $\mathcal{EL}$ knowledge bases (KBs) with complex role inclusions is PSPACE-hard and in PSPACE in certain cases; however, if complex role inclusions are restricted to role transitivity, a tight upper complexity bound has so far been unknown. Furthermore, the existing algorithms cannot handle reflexive roles, and they are not practicable. Finally, the problem is tractable for acyclic CQs and $\mathcal{ELH}$, and NP-complete for unrestricted CQs and $\mathcal{ELHO}$ KBs. In this paper we complete the complexity landscape of CQ answering for several important cases. In particular, we present a practicable NP algorithm for answering CQs over $\mathcal{ELHO}^s$ KBs—a logic containing all of OWL 2 EL, but with complex role inclusions restricted to role transitivity. Our preliminary evaluation suggests that the algorithm can be suitable for practical use. Moreover, we show that, even for a restricted class of so-called *arborescent* acyclic queries, CQ answering over $\mathcal{EL}$ KBs becomes NP-hard in the presence of either transitive or reflexive roles. Finally, we show that answering arborescent CQs over $\mathcal{ELHO}$ KBs is tractable, whereas answering acyclic CQs is NP-hard.

1 Introduction

Description logics (DLs) (Baader et al. 2007) are a family of knowledge representation languages that logically underpin the Web Ontology Language (OWL 2) (Cuenca Grau et al. 2008). DL knowledge bases (KBs) provide modern information systems with a flexible graph-like data model, and answering conjunctive queries (CQs) over such KBs is a core reasoning service in various applications (Calvanese et al. 2011). Thus, the investigation of the computational properties of CQ answering, as well as the development of practicable algorithms, have received a lot of attention lately.

For expressive DLs, CQ answering is at least exponential in *combined complexity* (Glimm et al. 2008; Ortiz, Rudolph, and Simkus 2011)—that is, measured in the combined size of the query and the KB. The problem is easier for the DL-Lite (Calvanese et al. 2007) and the $\mathcal{EL}$ (Baader, Brandt, and Lutz 2005) families of DLs, which logically underpin the QL and the EL profiles of OWL 2, respectively, and worst-case optimal, yet practicable algorithms are known (Kontchakov et al. 2011; Rodriguez-Muro, Kontchakov, and Zakharyaschev 2013; Eiter et al. 2012; Venetis, Stoilos, and Stamou 2014). One can reduce the complexity by restricting the query shape; for example, answering *acyclic* CQs (Yannakakis 1981) is tractable in relational databases. Bienvenu et al. (2013) have shown that answering acyclic CQs in DL-Lite$_{core}$ and $\mathcal{ELH}$ is tractable, whereas Gottlob et al. (2014) have shown it to be NP-hard in DL-Lite$_{\mathcal{R}}$.

In this paper, we consider answering CQs over KBs in the $\mathcal{EL}$ family of languages. No existing practical approach for $\mathcal{EL}$ supports complex role inclusions—a prominent feature of OWL 2 EL that can express complex properties of roles, including role transitivity. The known upper bound for answering CQs over $\mathcal{EL}$ KBs with complex role inclusions (Krötzsch, Rudolph, and Hitzler 2007) runs in PSPACE and uses automata techniques that are not practicable due to extensive don't-know nondeterminism. Moreover, this algorithm does not handle transitive roles specifically, but considers complex role inclusions. Hence, it is not clear whether the PSPACE upper bound is optimal in the presence of transitive roles only; this is interesting because role transitivity suffices to express simple graph properties such as reachability, and it is a known source of complexity of CQ answering (Eiter et al. 2009). Thus, to complete the landscape, we study the combined complexity of answering CQs over various extensions of $\mathcal{EL}$ and different classes of CQs. Our contributions can be summarised as follows.

In Section 3 we present a novel algorithm running in NP for answering CQs over $\mathcal{ELHO}^s$ KBs—a logic containing all of OWL 2 EL, but with complex role inclusions restricted to role transitivity—and thus settle the open question of the complexity for transitive and (locally) reflexive roles. Our procedure generalises the *combined approach with filtering* (Lutz et al. 2013) for $\mathcal{ELHO}$ by Stefanoni, Motik, and Horrocks (2013). We capture certain consequences of an $\mathcal{ELHO}^s$ KB by a datalog program; then, to answer a CQ, we evaluate the query over the datalog program to obtain *candidate answers*, and then we filter out unsound candidate answers. Transitive and reflexive roles, however, increase the complexity of the filtering step: unlike the filtering procedure for $\mathcal{ELHO}$, our filtering procedure runs in nondeterministic polynomial time, and we prove that this is worst-case optimal—that is, checking whether a candidate answer is sound is an NP-hard problem. To obtain a goal-directed filtering procedure, we developed optimisations that reduce the number of nondeterministic choices. Finally, our filtering procedure runs in NP only for candidate answers that depend on both the existential knowledge in the KB, and transitive or reflexive roles—that is, our algorithm exhibits pay-as-you-go behaviour. To evaluate the feasibility of our approach, we implemented a prototypical CQ answering system and we carried out a preliminary evaluation. Our results suggest that, although some queries may be challenging, our algorithm can be practicable in many cases.

In Section 4 we study the complexity of answering acyclic CQs over KBs expressed in various extensions of $\mathcal{EL}$. We introduce a new class of *arborescent* queries—tree-shaped acyclic CQs in which all roles point towards the parent. We prove that answering arborescent queries over $\mathcal{EL}$ KBs with either a single transitive role or a single reflexive role is NP-hard; this is interesting because Bienvenu et al. (2013) show that answering acyclic queries over $\mathcal{ELH}$ KBs is tractable, and it shows that our algorithm from Section 3 is optimal for arborescent (and thus also acyclic) queries. Moreover, we show that answering unrestricted acyclic CQs is NP-hard for $\mathcal{ELHO}$, but it becomes tractable for arborescent queries.

All proofs of our results are provided in a technical report (Stefanoni and Motik 2014).

2 Preliminaries

We use the standard notions of constants, (ground) terms, atoms, and formulas of first-order logic with the equality predicate $\approx$ (Fitting 1996); we assume that $\top$ and $\bot$ are unary predicates without any predefined meaning; and we often identify a conjunction with the set of its conjuncts. A substitution σ is a partial mapping of variables to terms; $\mathsf{dom}(\sigma)$ and $\mathsf{rng}(\sigma)$ are the domain and the range of σ, respectively; for convenience, we extend each σ to identity on ground terms; $\sigma|_S$ is the restriction of σ to a set of variables S; and, for α a term or a formula, $\sigma(\alpha)$ is the result of simultaneously replacing each free variable x occurring in α with $\sigma(x)$. Finally, $[i, j]$ is the set $\{i, i+1, \ldots, j-1, j\}$.

Rules and Conjunctive Queries An *existential rule* is a formula $\forall\vec{x}\forall\vec{y}.\varphi(\vec{x},\vec{y}) \rightarrow \exists\vec{z}.\psi(\vec{x},\vec{z})$ where φ and ψ are conjunctions of function-free atoms over variables $\vec{x}\cup\vec{y}$ and $\vec{x}\cup\vec{z}$, respectively. An *equality rule* is a formula of the form $\forall\vec{x}.\varphi(\vec{x}) \rightarrow s \approx t$ where φ is a conjunction of function-free atoms over variables $\vec{x}$, and s and t are function-free terms with variables in $\vec{x}$. A *rule base* Σ is a finite set of rules and function-free ground atoms; Σ is a *datalog program* if $\vec{z} = \emptyset$ for each existential rule in Σ. Please note that Σ is always satisfiable, as $\top$ and $\bot$ are ordinary unary predicates. We typically omit universal quantifiers in rules.

A *conjunctive query* (CQ) is a formula $q = \exists\vec{y}.\psi(\vec{x},\vec{y})$ where ψ is a conjunction of function-free atoms over variables $\vec{x}\cup\vec{y}$. Variables $\vec{x}$ are the *answer variables* of q. Let $N_V(q) = \vec{x}\cup\vec{y}$ and let $N_T(q)$ be the set of terms occurring in q. When $\vec{x}$ is empty, we call q a *Boolean CQ*.

For τ a substitution, let $\tau(q) = \exists\vec{z}.\tau(\psi)$, where $\vec{z}$ is obtained from $\vec{y}$ by removing each variable $y \in \vec{y}$ such that $\tau(y)$ is a constant, and by replacing each variable $y \in \vec{y}$ such that $\tau(y)$ is a variable with $\sigma(y)$.

Table 1: Translating $\mathcal{ELHO}^s$ Axioms into Rules

Type	Axiom		Rule
1	$A \sqsubseteq B$	$\rightsquigarrow$	$A(x) \rightarrow B(x)$
2	$A \sqsubseteq \{a\}$	$\rightsquigarrow$	$A(x) \rightarrow x \approx a$
3	$A_1 \sqcap A_2 \sqsubseteq A$	$\rightsquigarrow$	$A_1(x) \wedge A_2(x) \rightarrow A(x)$
4	$\exists R.A_1 \sqsubseteq A$	$\rightsquigarrow$	$R(x,y) \wedge A_1(y) \rightarrow A(x)$
5	$S \sqsubseteq R$	$\rightsquigarrow$	$S(x,y) \rightarrow R(x,y)$ $\mathsf{Self}_S(x) \rightarrow \mathsf{Self}_R(x)$
6	$\mathsf{range}(R, A)$	$\rightsquigarrow$	$R(x,y) \rightarrow A(y)$
7	$A_1 \sqsubseteq \exists R.A$	$\rightsquigarrow$	$A_1(x) \rightarrow \exists z.R(x,z) \wedge A(z)$
8	$\mathsf{trans}(R)$	$\rightsquigarrow$	$R(x,y) \wedge R(y,z) \rightarrow R(x,z)$
9	$\mathsf{refl}(R)$	$\rightsquigarrow$	$\top(x) \rightarrow R(x,x) \wedge \mathsf{Self}_R(x)$
10	$A \sqsubseteq \exists R.\mathsf{Self}$	$\rightsquigarrow$	$A(x) \rightarrow R(x,x) \wedge \mathsf{Self}_R(x)$
11	$\exists R.\mathsf{Self} \sqsubseteq A$	$\rightsquigarrow$	$\mathsf{Self}_R(x) \rightarrow A(x)$

Let Σ be a rule base and let $q = \exists\vec{y}.\psi(\vec{x},\vec{y})$ be a CQ over the predicates in Σ. A substitution π is a *certain answer* to q over Σ, written $\Sigma \models \pi(q)$, if $\mathsf{dom}(\pi) = \vec{x}$, each element of $\mathsf{rng}(\pi)$ is a constant, and $\mathcal{I} \models \pi(q)$ for each model $\mathcal{I}$ of Σ.

The DL $\mathcal{ELHO}^s$ is defined w.r.t. a signature consisting of mutually disjoint and countably infinite sets N_C, N_R, and N_I of *atomic concepts* (i.e., unary predicates), *roles* (i.e., binary predicates), and *individuals* (i.e., constants), respectively. We assume that $\top$ and $\bot$ do not occur in N_C. Each $\mathcal{ELHO}^s$ knowledge base can be *normalised* in polynomial time without affecting CQ answers (Krötzsch 2010), so we consider only normalised KBs. An $\mathcal{ELHO}^s$ *TBox* $\mathcal{T}$ is a finite set of axioms of the form shown in the left-hand side of Table 1, where $A_{(i)} \in N_C \cup \{\top\}$, $B \in N_C \cup \{\top, \bot\}$, $S, R \in N_R$, and $a \in N_I$; furthermore, TBox $\mathcal{T}$ is in $\mathcal{ELHO}$ if it contains only axioms of types 1–7. Relation $\sqsubseteq^*_{\mathcal{T}}$ is the smallest reflexive and transitive relation on the set of roles occurring in $\mathcal{T}$ such that $S \sqsubseteq^*_{\mathcal{T}} R$ for each $S \sqsubseteq R \in \mathcal{T}$. A role R is *simple* in $\mathcal{T}$ if $\mathsf{trans}(S) \notin \mathcal{T}$ for each $S \in N_R$ with $S \sqsubseteq^*_{\mathcal{T}} R$. An *ABox* $\mathcal{A}$ is a finite set of ground atoms constructed using the symbols from the signature. An $\mathcal{ELHO}^s$ *knowledge base* (KB) is a tuple $\mathcal{K} = \langle\mathcal{T}, \mathcal{A}\rangle$, where $\mathcal{T}$ is an $\mathcal{ELHO}^s$ TBox and an $\mathcal{A}$ is an ABox such that each role R occurring in axioms of types 10 or 11 in $\mathcal{T}$ is simple.

Let ind be a fresh atomic concept and, for each role R, let Self_R be a fresh atomic concept uniquely associated with R. Table 1 shows how to translate an $\mathcal{ELHO}^s$ TBox $\mathcal{T}$ into a rule base $\Xi_{\mathcal{T}}$. Furthermore, rule base $\mathsf{cls}_{\mathcal{K}}$ contains an atom $\mathsf{ind}(a)$ for each individual a occurring in $\mathcal{K}$, a rule $A(x) \rightarrow \top(x)$ for each atomic concept occurring in $\mathcal{K}$, and the following two rules for each role R occurring in $\mathcal{K}$.

$$\mathsf{ind}(x) \wedge R(x,x) \rightarrow \mathsf{Self}_R(x) \qquad (1)$$

$$R(x,y) \rightarrow \top(x) \wedge \top(y) \qquad (2)$$

For $\mathcal{K}$ a KB, let $\Xi_{\mathcal{K}} = \Xi_{\mathcal{T}} \cup \mathsf{cls}_{\mathcal{K}} \cup \mathcal{A}$; then, $\mathcal{K}$ is *unsatisfiable* iff $\Xi_{\mathcal{K}} \models \exists y.\bot(y)$. For q a CQ and π a substitution, we write $\mathcal{K} \models \pi(q)$ iff $\mathcal{K}$ is unsatisfiable or $\Xi_{\mathcal{K}} \models \pi(q)$. Our definition of the semantics of $\mathcal{ELHO}^s$ is unconventional, but equivalent to the usual one (Krötzsch 2010).

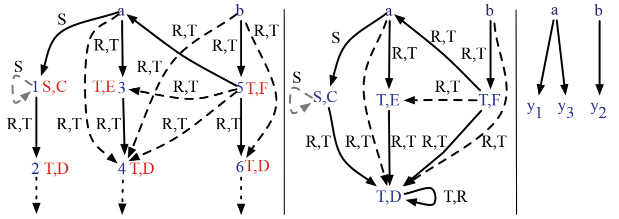

Figure 1: The models of $\Xi_\mathcal{K}$ and $\mathsf{D}_\mathcal{K}$, and the skeleton for q

3 Answering CQs over $\mathcal{ELHO}^s$ KBs

In this section, we present an algorithm for answering CQs over $\mathcal{ELHO}^s$ KBs running in NP. In the rest of this section, we fix $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ to be an arbitrary $\mathcal{ELHO}^s$ KB.

Certain answers to a CQ over $\Xi_\mathcal{K}$ can be computed by evaluating the CQ over a so-called *canonical model* that can be homomorphically embedded into all other models of $\Xi_\mathcal{K}$. It is well known (Krötzsch, Rudolph, and Hitzler 2007) that such models can be seen as a family of directed trees whose roots are the individuals occurring in $\Xi_\mathcal{K}$, and that contain three kinds edges: *direct edges* point from parents to children or to the individuals in $\Xi_\mathcal{K}$; *transitive edges* introduce shortcuts between these trees; and *self-edges* introduce loops on tree nodes. We call non-root elements *auxiliary*. Moreover, each auxiliary element can be uniquely associated with a rule of the form $A_1(x) \rightarrow \exists z.R(x, z) \wedge A(z)$ in $\Xi_\mathcal{K}$ that was used to generate it, and we cal R, A the element's *type*. Example 1 illustrates these observations.

Example 1. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be an $\mathcal{ELHO}^s$ KB whose $\mathcal{T}$ contains the following axioms and $\mathcal{A} = \{A(a), B(b)\}$.*

$$\begin{array}{lll}
A \sqsubseteq \exists S.C & E \sqsubseteq \exists T.D & G \sqsubseteq \{a\} \\
C \sqsubseteq \exists S.\mathsf{Self} & B \sqsubseteq \exists T.F & D \sqsubseteq \exists T.D \\
C \sqsubseteq \exists T.D & F \sqsubseteq \exists T.D & T \sqsubseteq R \\
A \sqsubseteq \exists T.E & F \sqsubseteq \exists T.G & \mathsf{trans}(T)
\end{array}$$

The left part of Figure 1 shows a canonical model I of $\Xi_\mathcal{K}$. Each auxiliary element is represented as a number showing the element's type. Transitive and self-edges are dashed, with the latter shown in grey. All other edges are solid, apart from the dotted edges that encode repetition of solid edges. Due to axiom $D \sqsubseteq \exists T.D$, model I is infinite.

A canonical model I of $\Xi_\mathcal{K}$ can be infinite, so a terminating CQ answering algorithm for $\mathcal{ELHO}^s$ cannot materialise I and evaluate CQs in it. Instead, we first show how to translate $\mathcal{K}$ into a datalog program $\mathsf{D}_\mathcal{K}$ that finitely captures the canonical model I of $\Xi_\mathcal{K}$; next, we present a CQ answering procedure that uses $\mathsf{D}_\mathcal{K}$ to answer CQs over $\Xi_\mathcal{K}$.

Datalog Translation

Krötzsch, Rudolph, and Hitzler (2008) translate $\mathcal{K}$ into datalog for the purposes of ontology classification, and Stefanoni, Motik, and Horrocks (2013) use this translation to answer CQs over $\mathcal{ELHO}$ KBs. Let $o_{R,A}$ be an *auxiliary individual* not occurring in N_I and uniquely associated with each role $R \in N_R$ and each atomic concept $A \in N_C \cup \{\top\}$ occurring in $\mathcal{K}$; intuitively, $o_{R,A}$ represent all auxiliary terms of type R, A in a canonical model I of $\Xi_\mathcal{K}$. We extend this translation by uniquely associating with each role R a *direct predicate* d_R to represent the direct edges in I.

Definition 1. *For each axiom $\alpha \in \mathcal{T}$ not of type 7, set $\mathsf{D}_\mathcal{T}$ contains the translation of α into a rule as shown in Table 1; moreover, for each axiom $A_1 \sqsubseteq \exists R.A \in \mathcal{T}$, set $\mathsf{D}_\mathcal{T}$ contains rule $A_1(x) \rightarrow R(x, o_{R,A}) \wedge \mathsf{d}_R(x, o_{R,A}) \wedge A(o_{R,A})$; finally, for each axiom $S \sqsubseteq R \in \mathcal{T}$, set $\mathsf{D}_\mathcal{T}$ contains rule $\mathsf{d}_S(x, y) \rightarrow \mathsf{d}_R(x, y)$. Then, $\mathsf{D}_\mathcal{K} = \mathsf{D}_\mathcal{T} \cup \mathsf{cls}_\mathcal{K} \cup \mathcal{A}$ is the* datalog program for $\mathcal{K}$.

Example 2. *The middle part of Figure 1 shows model J of the datalog program $\mathsf{D}_\mathcal{K}$ for the KB from Example 1. For clarity, auxiliary individuals $o_{R,A}$ are shown as R, A. Note that auxiliary individual $o_{T,G}$ is 'merged' in model J with individual a since $\mathsf{D}_\mathcal{K} \models o_{T,G} \approx a$. We use the notation from Example 1 to distinguish various kinds of edges.*

The following proposition shows how to use $\mathsf{D}_\mathcal{K}$ to test whether $\mathcal{K}$ is unsatisfiable.

Proposition 2. *$\mathcal{K}$ is unsatisfiable iff $\mathsf{D}_\mathcal{K} \models \exists y.\bot(y)$.*

The CQ Answering Algorithm

Program $\mathsf{D}_\mathcal{K}$ can be seen as a strengthening of $\Xi_\mathcal{K}$: all existential rules $A_1(x) \rightarrow \exists z.R(x, z) \wedge A(z)$ in $\Xi_\mathcal{K}$ are satisfied in a model J of $\mathsf{D}_\mathcal{K}$ using a single auxiliary individual $o_{R,A}$. Therefore, evaluating a CQ q in J produces a set of *candidate answers*, which provides us with an upper bound on the set of certain answers to q over $\Xi_\mathcal{K}$.

Definition 3. *A substitution τ is a* candidate answer *to a CQ $q = \exists \vec{y}.\psi(\vec{x}, \vec{y})$ over $\mathsf{D}_\mathcal{K}$ if $\mathsf{dom}(\tau) = N_V(q)$, each element of $\mathsf{rng}(\tau)$ is an individual occurring in $\mathsf{D}_\mathcal{K}$, and $\mathsf{D}_\mathcal{K} \models \tau(q)$. Such a candidate answer τ is* sound *if $\Xi_\mathcal{K} \models \tau|_{\vec{x}}(q)$.*

Stefanoni, Motik, and Horrocks (2013) presented a filtering step that removes unsound candidate answers; however, Example 3 shows that this step can be incomplete when the query contains roles that are not simple.

Example 3. *Let $\mathcal{K}$ be as in Example 1 and let*

$$q = \exists y.A(x_1) \wedge R(x_1, y) \wedge B(x_2) \wedge R(x_2, y) \wedge D(y).$$

Moreover, let π be the substitution such that $\pi(x_1) = a$ and $\pi(x_2) = b$, and let τ be such that $\pi \subseteq \tau$ and $\tau(y) = o_{T,D}$. Using models I and J from Figure 1, one can easily see that $\Xi_\mathcal{K} \models \pi(q)$ and $\mathsf{D}_\mathcal{K} \models \tau(q)$. However, q contains a 'fork' $R(x_1, y) \wedge R(x_2, y)$, and τ maps y to an auxiliary individual, so this answer is wrongly filtered as unsound.

Algorithm 1 specifies a procedure $\mathsf{isSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ that checks whether a candidate answer is sound. We discuss the intuitions using the KB from Example 1, and the query q and the candidate answer τ from Example 4.

Example 4. *Let q and τ be as follows. Using Figure 1, one can easily see that $\mathsf{D}_\mathcal{K} \models \tau(q)$.*

$$\begin{aligned} q = \exists \vec{y}.\ & S(x, y_1) \wedge S(y_1, y_1) \wedge R(x, y_3) \wedge D(y_3) \wedge \\ & R(y_2, y_3) \wedge F(y_2) \wedge T(y_2, x) \end{aligned}$$

$$\tau = \{x \mapsto a,\ y_1 \mapsto o_{S,C},\ y_2 \mapsto o_{T,F},\ y_3 \mapsto o_{T,D}\}$$

We next show how $\mathsf{isSound}(q, \mathsf{D}_{\mathcal{K}}, \tau)$ decides that τ is sound—that is, that a substitution π mapping the variables in q to terms in I exists such that $\pi(x) = a$ and $\pi(q) \subseteq I$. Substitution τ already provides us with some constraints on π: it must map variable y_1 to 1 and variable y_2 to 5, since these are the only elements of I of types S, C and T, F, respectively. In contrast, substitution π can map variable y_3 to either one of 2, 4, and 6. Each such substitution π is guaranteed to satisfy all unary atoms of q, all binary atoms of q that τ maps to direct edges pointing towards (non-auxiliary) individuals from N_I, and all binary atoms of q that contain a single variable and that τ maps to the self-edge in J. Atoms $T(y_2, x)$ and $S(y_1, y_1)$ in q satisfy these conditions, and so we call them *good* w.r.t. τ. To show that τ is sound, we must demonstrate that all other atoms of q are satisfied.

Step 1 of Algorithm 1 implements the 'fork' and 'aux-acyclicity' checks (Stefanoni, Motik, and Horrocks 2013). To guarantee completeness, we consider only those binary atoms in q that contain simple roles, and that τ maps onto direct edges in J pointing towards auxiliary elements. We call these atoms *aux-simple* as they can be mapped onto the direct edges in I pointing towards auxiliary elements. In step 1 we compute a new query $q_\sim$ by applying all constraints derived by the fork rule, and in the rest of the algorithm we consider $q_\sim$ instead of q. In our example, atom $S(x, y_1)$ is the only aux-simple atom, so q does not contain forks and $q_\sim = q$. When all binary atoms occurring in q are good or aux-simple, step 1 guarantees that τ is sound. Query q from Example 4, however, contains binary atoms that are neither good nor aux-simple, so we proceed to step 3.

Next, in step 3 we guess a renaming σ for the variables in $q_\sim$ to take into account that distinct variables in $q_\sim$ that τ maps to the same auxiliary individual can be mapped to the same auxiliary element of I, and so in the rest of Algorithm 1 we consider $\sigma(q_\sim)$ instead of $q_\sim$. In our example, we guess σ to be identity, so $\sigma(q_\sim) = q_\sim = q$.

In step 4, we guess a *skeleton* for $\sigma(q_\sim)$, which is a finite structure that finitely describes the (possibly infinite) set of all substitutions π mapping the variables in $\sigma(q_\sim)$ to distinct auxiliary elements of I. The right part of Figure 1 shows the skeleton $\mathcal{S}$ for our example query. The vertices of $\mathcal{S}$ are the (non-auxiliary) individuals from $\mathsf{D}_{\mathcal{K}}$ and the variables from $\sigma(q_\sim)$ that τ maps to auxiliary individuals, and they are arranged into a forest rooted in N_I. Such $\mathcal{S}$ represents those substitutions π that map variables y_1 and y_3 to auxiliary elements of I under individual a, and that map variable y_2 to an auxiliary element of I under individual b.

In steps 5–15, our algorithm labels each edge $\langle v', v\rangle \in \mathcal{S}$ with a set of roles $L(v', v)$; after these steps, $\mathcal{S}$ represents those substitutions π that satisfy the following property (E):

> for each role $P \in L(v', v)$, a path from $\tau(v')$ to $\tau(v)$ in J exists that consists only of direct edges labelled by role P pointing to auxiliary individuals.

We next show how atoms of $\sigma(q_\sim)$ that are not good contribute to the labelling of $\mathcal{S}$. Atom $S(x, y_1)$ is used in step 6 to label edge $\langle a, y_1\rangle$. For atom $R(x, y_3)$, in step 8 we let $P = T$ and we label edge $\langle a, y_3\rangle$ with P. Using Figure 1 and the axioms in Example 1, one can easily check that the conditions in steps 8 and 9 are satisfied. For atom $R(y_2, y_3)$, variables y_2 and y_3 are not reachable in $\mathcal{S}$, so we must split the path from y_2 to y_3. Thus, in step 8 we let $P = T$, and in step 13 we let $a_t = a$; hence, atom $R(y_2, y_3)$ is split into atoms $T(y_2, a)$ and $T(a, y_3)$. The former is used in step 14 to check that a direct path exists in J connecting $o_{T,F}$ with a, and the latter is used to label edge $\langle a, y_2\rangle$.

After the for-loop in steps 7–15, skeleton $\mathcal{S}$ represents all substitutions satisfying (E). In step 17, function exist exploits the direct predicates from $\mathsf{D}_{\mathcal{K}}$ to find the required direct paths in J, thus checking whether at least one such substitution exists (see Definition 11). Using Figure 1, one can check that substitution π where $\pi(y_3) = 4$ and that maps all other variables as stated above is the only substitution satisfying the constraints imposed by $\mathcal{S}$; hence, $\mathsf{isSound}$ returns t, indicating that candidate answer τ is sound.

We now formalise the intuitions that we have just presented. Towards this goal, in the rest of this section we fix a CQ q' and a candidate answer τ' to q' over $\mathsf{D}_{\mathcal{K}}$.

Due to equality rules, auxiliary individuals in $\mathsf{D}_{\mathcal{K}}$ may be equal to individuals from N_I, thus not representing auxiliary elements of I. Hence, set $\mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ in Definition 4 provides us with all auxiliary individuals that are not equal to an individual from N_I. Moreover, to avoid dealing with equal individuals, we replace in query q' all terms that τ' does not map to individuals in $\mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ with a single canonical representative, and we do analogously for τ'; this replacement produces CQ q and substitution τ. Since q and τ are obtained by replacing equals by equals, we have $\mathsf{D}_{\mathcal{K}} \models \tau(q)$. Our filtering procedure uses q and τ to check whether τ' is sound.

Definition 4. *Let $>$ be a total order on ground terms such that $o_{R,A} > a$ for all individuals $o_{R,A}$ and $a \in N_I$ from $\mathsf{D}_{\mathcal{K}}$. Set $\mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ contains each individual u from $\mathsf{D}_{\mathcal{K}}$ for which no individual $a \in N_I$ exists such that $\mathsf{D}_{\mathcal{K}} \models u \approx a$. For each individual u from $\mathsf{D}_{\mathcal{K}}$, let $u_\approx = u$ if $u \in \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$; otherwise, let $u_\approx$ be the smallest individual $a \in N_I$ in the ordering $>$ such that $\mathsf{D}_{\mathcal{K}} \models u \approx a$. Set $\mathsf{ind}_{\mathsf{D}_{\mathcal{K}}}$ contains $a_\approx$ for each individual $a \in N_I$ occurring in $\mathsf{D}_{\mathcal{K}}$. Then query q is obtained from q' by replacing each term $t \in N_T(q')$ such that $\tau'(t) \notin \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ with $\tau'(t)_\approx$; substitution τ is obtained by restricting τ' to only those variables occurring in q.*

Next, we define good and aux-simple atoms w.r.t. τ.

Definition 5. *Let $R(s, t)$ be an atom where $\tau(s)$ and $\tau(t)$ are defined. Then, $R(s, t)$ is* good *if $\tau(t) \in N_I$, or $s = t$ and $\mathsf{D}_{\mathcal{K}} \models \mathsf{Self}_R(\tau(s))$. Furthermore, $R(s, t)$ is* aux-simple *if $s \neq t$, R is a simple role, $\tau(t) \in \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$, and $\tau(s) = \tau(t)$ implies $\mathsf{D}_{\mathcal{K}} \not\models \mathsf{Self}_R(\tau(s))$.*

Note that, if $R(s, t)$ is not good, then t is a variable and $\tau(t) \in \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$. Moreover, by the definition of $\mathsf{D}_{\mathcal{K}}$, if atom $R(s, t)$ is aux-simple, then $\mathsf{D}_{\mathcal{K}} \models \mathsf{d}_R(\tau(s), \tau(t))$. The following definition introduces the query $q_\sim$ obtained by applying the fork rule by Stefanoni, Motik, and Horrocks (2013) to only those atoms that are aux-simple.

Definition 6. *Relation $\sim \subseteq N_T(q) \times N_T(q)$ for q and τ is the smallest reflexive, symmetric, and transitive relation closed under the* fork *rule.*

$$(\mathsf{fork})\ \frac{s' \sim t'}{s \sim t}\ \ \begin{array}{l}\textit{R(s, s') and P(t, t') are aux-simple}\\ \textit{atoms in q w.r.t. } \tau\end{array}$$

Query $q_\sim$ is obtained from query q by replacing each term $t \in N_T(q)$ with an arbitrary, but fixed representative of the equivalence class of $\sim$ that contains t.

To check whether $q_\sim$ is aux-acyclic, we next introduce the *connection graph* cg for q and τ that contains a set E_s of edges $\langle v', v\rangle$ for each aux-simple atom $R(v', v) \in q_\sim$. In addition, cg also contains a set E_t of edges $\langle v', v\rangle$ that we later use to guess a skeleton for $\sigma(q_\sim)$ more efficiently. By the definition of aux-simple atoms, we have $E_s \subseteq E_t$.

Definition 7. *The* connection graph for q and τ *is a triple* $\mathsf{cg} = \langle V, E_s, E_t\rangle$ *where $E_s, E_t \subseteq V \times V$ are smallest sets satisfying the following conditions.*

- $V = \mathsf{ind}_{\mathsf{D}_\mathcal{K}} \cup \{z \in N_V(q_\sim) \mid \tau(z) \in \mathsf{aux}_{\mathsf{D}_\mathcal{K}}\}$.
- *Set E_s contains $\langle v', v\rangle$ for all $v', v \in V$ for which a role R exist such that $R(v', v)$ is an aux-simple atom in $q_\sim$.*
- *Set E_t contains $\langle v', v\rangle$ for all $v', v \in V$ such that individuals $\{u_1, \ldots, u_n\} \subseteq \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ and roles $R_1, \ldots, R_n$ exist with $n > 0$, $u_n = \tau(v)$, and $\mathsf{D}_\mathcal{K} \models \mathsf{d}_{R_i}(u_{i-1}, u_i)$ for each $i \in [1, n]$ and $u_0 = \tau(v')$.*

Function $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ from Definition 8 ensures that τ satisfies the constraints in $\sim$, and that $q_\sim$ does not contain cycles consisting only of aux-simple atoms.

Definition 8. *Function* $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ *returns* t *if and only if the two following conditions hold.*

1. *For all $s, t \in N_T(q)$, if $s \sim t$, then $\tau(s) = \tau(t)$.*
2. *$\langle V, E_s\rangle$ is a directed acyclic graph.*

We next define the notions of a variable renaming for q and τ, and of a skeleton for q and σ.

Definition 9. *A substitution σ with* $\mathsf{dom}(\sigma) = V \cap N_V(q)$ *and* $\mathsf{rng}(\sigma) \subseteq \mathsf{dom}(\sigma)$ *is a* variable renaming for q and τ *if*

1. *for each $v \in \mathsf{dom}(\sigma)$, we have $\tau(v) = \tau(\sigma(v))$,*
2. *for each $v \in \mathsf{rng}(\sigma)$, we have $\sigma(v) = v$, and*
3. *directed graph $\langle \sigma(V), \sigma(E_s)\rangle$ is a forest.*

Definition 10. *A* skeleton *for q and a variable renaming σ is a directed graph $\mathcal{S} = \langle \mathcal{V}, \mathcal{E}\rangle$ where $\mathcal{V} = \sigma(V)$, and $\mathcal{E}$ satisfies $\sigma(E_s) \subseteq \mathcal{E} \subseteq \sigma(E_t)$ and it is a forest whose roots are the individuals occurring in $\mathcal{V}$.*

Finally, we present function exist that checks whether one can satisfy the constraints imposed by the roles $L(v', v)$ labelling a skeleton edge $\langle v', v\rangle \in \mathcal{E}$.

Definition 11. *Given individuals u' and u, and a set of roles L, function* $\mathsf{exist}(u', u, L)$ *returns* t *if and only if individuals $\{u_1, \ldots, u_n\} \subseteq \mathsf{aux}_{\mathsf{D}_\mathcal{K}}$ with $n > 0$ and $u_n = u$ exist where*

- *if $S \in L$ exists such that $\mathsf{trans}(S) \notin \mathcal{T}$, then $n = 1$; and*
- *$u_0 = u'$, and $\mathsf{D}_\mathcal{K} \models \mathsf{d}_R(u_{i-1}, u_i)$ for each $R \in L$ and each $i \in [1, n]$.*

Candidate answer τ' for q' over $\mathsf{D}_\mathcal{K}$ is sound, if the nondeterministic procedure $\mathsf{isSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ from Algorithm 1 returns t, as shown by Theorem 12.

Algorithm 1: $\mathsf{isSound}(q, \mathsf{D}_\mathcal{K}, \tau)$

```
if $\mathsf{isDSound}(q, \mathsf{D}_\mathcal{K}, \tau) = \mathsf{f}$ then return $\mathsf{f}$
return $\mathsf{t}$ if each $R(s, t) \in q_\sim$ is good or aux-simple
guess a variable renaming $\sigma$ for $q$ and $\tau$
guess a skeleton $\mathcal{S} = \langle \mathcal{V}, \mathcal{E}\rangle$ for $q$, $\sigma$, and $\tau$
for $\langle v', v\rangle \in \mathcal{E}$, let $L(v', v) = \emptyset$
for aux-simple atom $R(s, t) \in \sigma(q_\sim)$, add $R$ to $L(s, t)$
for neither good nor aux-simple $R(s, t) \in \sigma(q_\sim)$ do
    guess role $P$ s.t. $\mathsf{D}_\mathcal{K} \models P(\tau(s), \tau(t))$ and $P \sqsubseteq^*_\mathcal{T} R$
    if $\langle s, t\rangle \notin \mathcal{E}$ and $\mathsf{trans}(P) \notin \mathcal{T}$ then return $\mathsf{f}$
    if $s$ reaches $t$ in $\mathcal{E}$ then
        let $v_0, \ldots, v_n$ be the path from $s$ to $t$ in $\mathcal{E}$
    else
        let $a_t$ be the root reaching $t$ in $\mathcal{E}$ via $v_0, \ldots, v_n$
        if $\mathsf{D}_\mathcal{K} \not\models P(\tau(s), a_t)$ then return $\mathsf{f}$
    for $i \in [1, n]$, add $P$ to $L(v_{i-1}, v_i)$
for $\langle v', v\rangle \in \mathcal{E}$ do
    if $\mathsf{exist}(\tau(v'), \tau(v), L(v', v)) = \mathsf{f}$ then return $\mathsf{f}$
return $\mathsf{t}$
```

Theorem 12. *Let π' be a substitution. Then $\Xi_\mathcal{K} \models \pi'(q')$ iff $\mathcal{K}$ is unsatisfiable, or a candidate answer τ' to q' over $\mathsf{D}_\mathcal{K}$ exists such that $\tau'|_{\vec{x}} = \pi'$ and the following conditions hold:*

1. *for each $x \in \vec{x}$, we have $\tau'(x) \in N_I$, and*
2. *a nondeterministic computation exists such that function* $\mathsf{isSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ *returns* t.

The following results show that our function isSound runs in nondeterministic polynomial time.

Theorem 13. *Function* $\mathsf{isSound}(q, \mathsf{D}_\mathcal{K}, \tau)$ *can be implemented so that*

1. *it runs in nondeterministic polynomial time,*
2. *if each binary atom in q is either good or aux-simple w.r.t. τ, it runs in polynomial time, and*
3. *if the TBox $\mathcal{T}$ and the query q are fixed, it runs in polynomial time in the size of the ABox $\mathcal{A}$.*

Each rule in $\mathsf{D}_\mathcal{K}$ contains a fixed number of variables, so we can compute all consequences of $\mathsf{D}_\mathcal{K}$ using polynomial time. Thus, we can compute CQ q and substitution τ in polynomial time, and by Proposition 2, we can also check whether $\mathcal{K}$ is unsatisfiable using polynomial time; hence, by Theorem 13, we can check whether a certain answer to q' over $\Xi_\mathcal{K}$ exists using nondeterministic polynomial time in combined complexity (i.e., when the ABox, the TBox, and the query are all part of the input), and in polynomial time in data complexity (i.e., when the TBox and the query are fixed, and only the ABox is part of the input).

The filtering procedure by Stefanoni, Motik, and Horrocks (2013) is polynomial, whereas the one presented in this paper introduces a source of intractability. In Theorem 14 we show that checking whether a candidate answer is sound is an NP-hard problem; hence, this complexity increase is unavoidable. We prove our claim by reducing the NP-hard problem of checking satisfiability of a 3CNF formula φ (Garey and Johnson 1979). Towards this goal, we define an $\mathcal{ELHO}^s$ KB $\mathcal{K}_\varphi$ and a Boolean CQ q_φ such that φ is satisfiable if and only if $\Xi_{\mathcal{K}_\varphi} \models q_\varphi$. Furthermore, we define a substitution τ_φ, and we finally show that τ_φ is a unique candidate answer to q_φ over $\mathsf{D}_{\mathcal{K}_\varphi}$.

Theorem 14. *Checking whether a candidate answer is sound is* NP*-hard.*

		Inds.	Unary atoms	Binary atoms	Total atoms	Ratio
U5	before	100,848	169,079	296,941	466,020	
	after	100,873	511,115	1,343,848	1,854,963	3.98
U10	before	202,387	339,746	598,695	938,441	
	after	202,412	1,026,001	2,714,214	3,740,215	3.98
U20	before	426,144	714,692	1,259,936	1,974,628	
	after	426,169	2,157,172	5,720,670	7,877,842	3.99

Table 2: Evaluation results

	q_3^l				q_1^t				q_2^t				q_3^t				q_4^t			
	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N	C	U	F	N
U5	10	0	0.06	0	73K	12	1.71	7.55	3K	0	0.01	0	157K	66	1.07	8.6	30K	63	2.44	10.9
U10	22	0	0.06	0	149K	12	1.68	7.54	6K	0	0.01	0	603K	81	1.20	9.6	61K	63	2.44	10.9
U20	43	0	0.07	0	313K	12	1.66	7.55	12K	0	0.01	0	2.6M	90	1.28	10.3	129K	63	2.44	10.9

(C) total number of candidate answers (U) percentage of unsound answers (F) average filtering time in ms (N) average number of nondeterministic choices required for each candidate answer

Preliminary Evaluation

We implemented our algorithm in a prototypical system, and we conducted a preliminary evaluation with the goal of showing that the number of consequences of $\mathsf{D}_{\mathcal{K}}$ is reasonably small, and that the nondeterminism of the filtering procedure is manageable. Our prototype uses the RDFox (Motik et al. 2014) system to materialise the consequences of $\mathsf{D}_{\mathcal{K}}$. We ran our tests on a MacBook Pro with 4GB of RAM and a 2.4Ghz Intel Core 2 Duo processor.

We tested our system using the version of the LSTW benchmark (Lutz et al. 2013) by Stefanoni, Motik, and Horrocks (2013). The TBox of the latter is in $\mathcal{ELHO}$, and we extended it to $\mathcal{ELHO}^s$ by making the role *subOganizationOf* transitive and by adding an axiom of type 5 and an axiom of type 7. We used the data generator provided by LSTW to generate KBs U5, U10, and U20 of 5, 10, and 20 universities, respectively. Finally, only query q_3^l from the LSTW benchmark uses transitive roles, so we have manually created four additional queries. Our system, the test data, and the queries are all available online.[1] We evaluated the practicality of our approach using the following two experiments.

First, we compared the size of the materialised consequences of $\mathsf{D}_{\mathcal{K}}$ with that of the input data. As the left-hand side of Table 2 shows, the ratio between the two is four, which, we believe, is acceptable in most practical scenarios.

Second, we measured the 'practical hardness' of our filtering step on our test queries. As the right-hand side of Table 2 shows, soundness of a candidate answer can typically be tested in as few as several milliseconds, and the test involves a manageable number of nondeterministic choices. Queries q_3^t and q_4^t were designed to obtain a lot of candidate answers with auxiliary individuals, so they retrieve many unsound answers. However, apart from query q_3^t, the percentage of the candidate answers that turned out to be unsound does not change with the increase in the size of the ABox. Therefore, while some queries may be challenging, we believe that our algorithm can be practicable in many cases.

4 Acyclic and Arborescent Queries

In this section, we prove that answering a simple class of tree-shaped acyclic CQs—which we call *arborescent*—over $\mathcal{ELHO}$ KBs is tractable, whereas answering acyclic queries is NP-hard. In addition, we show that extending $\mathcal{EL}$ with transitive or reflexive roles makes answering arborescent queries NP-hard. This is in contrast with the recent result by Bienvenu et al. (2013), who show that answering acyclic CQs over $\mathcal{ELH}$ KBs is tractable. We start by introducing acyclic and arborescent queries.

Definition 15. *For q a Boolean CQ, $\mathsf{dg}_q = \langle N_V(q), E\rangle$ is a directed graph where $\langle x, y\rangle \in E$ for each $R(x, y) \in q$. Query q is* acyclic *if the graph obtained from dg_q by removing the orientation of edges is acyclic; q is* arborescent *if q contains no individuals and dg_q is a rooted tree with all edges pointing towards the root.*

Definition 16 and Theorem 17 show how to answer arborescent CQs over $\mathcal{ELHO}$ KBs in polynomial time. Intuitively, we apply the fork rule (cf. Definition 6) bottom-up, starting with the leaves of q and spread constraints upwards.

Definition 16. *Let $\mathcal{K}$ be an $\mathcal{ELHO}$ KB, let $\mathsf{D}_{\mathcal{K}}$ be the datalog program for $\mathcal{K}$, let $\mathsf{ind}_{\mathsf{D}_{\mathcal{K}}}$ and $\mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$ be as specified in Definition 4, and let q be an arborescent query rooted in $r \in N_V(q)$. For each $y \in N_V(q)$ with $y \neq r$, and each $V \subseteq N_V(q)$, sets r_y and P_V are defined as follows.*

$$\mathsf{r}_y = \{R \in N_R \mid R(y, x) \in q \textit{ with } x \textit{ the parent of } y \textit{ in } \mathsf{dg}_q\}$$

$$\mathsf{P}_V = \{y \in N_V(q) \mid \exists x \in V \textit{ with } x \textit{ the parent of } y \textit{ in } \mathsf{dg}_q\}$$

Set RT is the smallest set satisfying the following conditions.

- *$\{r\} \in \mathsf{RT}$ and the level of $\{r\}$ is 0.*
- *For each set $V \in \mathsf{RT}$ with level n, we have $\mathsf{P}_V \in \mathsf{RT}$ and the level of P_V is $n + 1$.*
- *For each set $V \in \mathsf{RT}$ with level n and each $y \in \mathsf{P}_V$, we have $\{y\} \in \mathsf{RT}$ and the level of $\{y\}$ is $n + 1$.*

For each $V \in \mathsf{RT}$, set c_V contains each $u \in \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}} \cup \mathsf{ind}_{\mathsf{D}_{\mathcal{K}}}$ such that $\mathsf{D}_{\mathcal{K}} \models B(u)$ for each unary atom $B(x) \in q$ with $x \in V$. By reverse-induction on the level of the sets in RT, each $V \in \mathsf{RT}$ is associated with a set $\mathsf{A}_V \subseteq \mathsf{ind}_{\mathsf{D}_{\mathcal{K}}} \cup \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}}$.

- *For each set $V \in \mathsf{RT}$ of maximal level, let $\mathsf{A}_V = \mathsf{c}_V$.*
- *For $V \in \mathsf{RT}$ a set of level n where A_V is undefined but A_W has been defined for each $W \in \mathsf{RT}$ of level $n + 1$, let $\mathsf{A}_V = \mathsf{c}_V \cap (\mathsf{i}_V \cup \mathsf{a}_V)$, where i_V and a_V are as follows.*

$$\mathsf{i}_V = \{u \in \mathsf{ind}_{\mathsf{D}_{\mathcal{K}}} \mid \forall y \in \mathsf{P}_V \exists u' \in \mathsf{A}_{\{y\}}.\mathsf{D}_{\mathcal{K}} \models \bigwedge_{R \in \mathsf{r}_y} R(u', u)\}$$

$$\mathsf{a}_V = \{u \in \mathsf{aux}_{\mathsf{D}_{\mathcal{K}}} \mid \exists u' \in \mathsf{A}_{\mathsf{P}_V} \forall y \in \mathsf{P}_V.\mathsf{D}_{\mathcal{K}} \models \bigwedge_{R \in \mathsf{r}_y} \mathsf{d}_R(u', u)\}$$

Function $\mathsf{entails}(\mathsf{D}_{\mathcal{K}}, q)$ *returns* t *if and only if* $\mathsf{A}_{\{r\}}$ *is nonempty.*

Theorem 17. *For $\mathcal{K}$ a satisfiable $\mathcal{ELHO}$ KB and q an arborescent query, function* $\mathsf{entails}(\mathsf{D}_{\mathcal{K}}, q)$ *returns* t *if and only if $\Xi_{\mathcal{K}} \models q$. Furthermore, function* $\mathsf{entails}(\mathsf{D}_{\mathcal{K}}, q)$ *runs in time polynomial in the input size.*

[1] http://www.cs.ox.ac.uk/isg/tools/EOLO/

Finally, we show that (unless PTIME = NP), answering arbitrary acyclic queries over $\mathcal{ELHO}$ KBs is harder than answering arborescent queries, and we show that adding transitive or reflexive roles to the DL $\mathcal{EL}$ makes answering arborescent queries intractable.

Theorem 18. *For $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ a KB and q a Boolean CQ, checking $\mathcal{K} \models q$ is* NP*-hard in each of the following cases.*

1. *The query q is acyclic and the TBox $\mathcal{T}$ is in $\mathcal{ELHO}$.*
2. *The query q is arborescent and the TBox $\mathcal{T}$ consists only of axioms of type* 1 *and* 7*, and of one axiom of type* 8*.*
3. *The query q is arborescent and the TBox $\mathcal{T}$ consists only of axioms of type* 1 *and* 7*, and of one axiom of type* 9*.*

5 Outlook

In future, we shall adapt our filtering procedure to detect unsound answers already during query evaluation. Moreover, we shall extend Algorithm 1 to handle complex role inclusions, thus obtaining a practicable approach for OWL 2 EL.

Acknowledgements

This work was supported by Alcatel-Lucent; the EU FP7 project OPTIQUE; and the EPSRC projects MASI^3, Score!, and DBOnto.

References

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2007. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the $\mathcal{EL}$ envelope. In Kaelbling, L. P., and Saffiotti, A., eds., *IJCAI 2005*, 364–369.

Bienvenu, M.; Ortiz, M.; Simkus, M.; and Xiao, G. 2013. Tractable queries for lightweight description logics. In *IJCAI 2013*.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. of Automated Reasoning* 39(3):385–429.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; Rodriguez-Muro, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2011. The Mastro system for ontology-based data access. *Semantic Web Journal* 2(1):43–53.

Cuenca Grau, B.; Horrocks, I.; Motik, B.; Parsia, B.; Patel-Schneider, P.; and Sattler, U. 2008. OWL 2: The next step for OWL. *Journal of Web Semantics* 6(4):309–322.

Eiter, T.; Lutz, C.; Ortiz, M.; and Simkus, M. 2009. Query answering in description logics with transitive roles. In *IJCAI 2009*, 759–764.

Eiter, T.; Ortiz, M.; Simkus, M.; Tran, T.-K.; and Xiao, G. 2012. Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In *AAAI 2012*.

Fitting, M. 1996. *First-order logic and automated theorem proving (2nd ed.)*. Springer-Verlag New York, Inc.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.

Glimm, B.; Horrocks, I.; Lutz, C.; and Sattler, U. 2008. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. *Journal of Artif. Intell. Res.* 31:151–198.

Gottlob, G.; Kikot, S.; Kontchakov, R.; Podolskii, V. V.; Schwentick, T.; and Zakharyaschev, M. 2014. The price of query rewriting in ontology-based data access. *Artif. Intell.* 213:42–59.

Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Zakharyaschev, M. 2011. The combined approach to ontology-based data access. In Walsh, T., ed., *IJCAI 2011*, 2656–2661.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2007. Conjunctive queries for a tractable fragment of OWL 1.1. In Aberer, K., et al., eds., *ISWC 2007*, 310–323.

Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2008. ELP: Tractable rules for OWL 2. In Sheth, A., et al., eds., *ISWC 2008*, 649–664.

Krötzsch, M. 2010. Efficient inferencing for OWL EL. In *JELIA 2010*, volume 6341, 234–246.

Lutz, C.; Seylan, I.; Toman, D.; and Wolter, F. 2013. The combined approach to OBDA: Taming role hierarchies using filters. In *ISWC 2013*, volume 8218, 314–330.

Motik, B.; Nenov, Y.; Piro, R.; Horrocks, I.; and Olteanu, D. 2014. Parallel materialisation of datalog programs in centralised, main-memory RDF systems. In *AAAI 2014*.

Ortiz, M.; Rudolph, S.; and Simkus, M. 2011. Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In Walsh, T., ed., *IJCAI 2011*, 1039–1044.

Rodriguez-Muro, M.; Kontchakov, R.; and Zakharyaschev, M. 2013. Ontology-based data access: Ontop of databases. In *ISWC 2013*, 558–573. Springer.

Stefanoni, G., and Motik, B. 2014. Answering conjunctive queries over $\mathcal{EL}$ knowledge bases with transitive and reflexive roles. *CoRR* abs/1411.2516.

Stefanoni, G.; Motik, B.; and Horrocks, I. 2013. Introducing nominals to the combined query answering approaches for EL. In *AAAI 2013*.

Venetis, T.; Stoilos, G.; and Stamou, G. B. 2014. Query extensions and incremental query rewriting for OWL 2 QL ontologies. *J. Data Semantics* 3(1):1–23.

Yannakakis, M. 1981. Algorithms for acyclic database schemes. In *VLDB 1981*, 82–94.