Conjunctive Query Answering in $\mathcal{SH}$ using Knots [⋆]

Magdalena Ortiz, Mantas Šimkus, and Thomas Eiter

Institut für Informationssysteme 184/3, Technische Universität Wien
Favoritenstraße 9-11, A-1040 Vienna, Austria
(ortiz|simkus|eiter)@kr.tuwien.ac.at

Abstract. Answering conjunctive queries (CQs) has been recognized as a key task for the usage of Description Logics (DLs) in a number of applications. The problem has been studied by many authors, who developed a number of different techniques for it. We present a novel method for CQ answering based on knots, which are schematic trees of depth ≤ 1. It yields an algorithm for CQ answering that works in exponential time for $\mathcal{ALCH}$ and for large classes of CQs in $\mathcal{SH}$. This improves over previous algorithms which require double exponential time and is worst-case optimal, as already satisfiability testing in $\mathcal{ALC}$ is EXPTIME-complete. Our result reconfirms Lutz's result that inverse roles cause an exponential jump in complexity, being the problem 2EXPTIME-complete for $\mathcal{ALCI}$. The algorithm is coNP, and hence also worst-case optimal, under data complexity.

1 Introduction

In the last years, Description Logics (DLs) have increasingly received attention as formalisms to represent richer domain models in various contexts, including the Semantic Web, data and information integration, peer-to-peer data management, and ontology-based data access. The wider use of DLs also raises the need for reasoning services beyond traditional satisfiability, subsumption, and instance checking. In particular, answering conjunctive queries (CQs) over knowledge bases has been recognized as a key task in this respect and studied in many papers, including [12, 6, 4, 5, 9, 2, 1, 3, 7, 11, 14].

Answering CQs in expressive DLs containing $\mathcal{ALC}$, like $\mathcal{SHIQ}$, $\mathcal{SRIQ}$ and $\mathcal{DLR}$, is at least EXPTIME-hard, since it subsumes the satisfiability problem of $\mathcal{ALC}$ knowledge bases which is well-known to be EXPTIME-complete. An important result on the computational complexity of CQ answering in expressive DLs was shown by Lutz, who proved that it is 2EXPTIME-hard for all DLs containing $\mathcal{ALCI}$ [9]; thus for the aforementioned DLs, corresponding 2EXPTIME upper bounds from [2, 3, 6, 4] are tight. Furthermore, Lutz identified inverse roles as the source of this exponential jump in complexity, and reported in [9] that the problem is in EXPTIME for $\mathcal{ALC}$. Using techniques similar to [4], he gave an EXPTIME algorithm for answering CQs in $\mathcal{ALCHQ}$; see [10].

Various approaches for answering CQs in expressive DLs have been used; they range from adapted tableaux procedures [8, 12, 11] over incorporating the query into the knowledge base [2, 17, 4, 5] and resolution-based techniques [6] to automata-based algorithms [3]. In this paper, we consider a novel method. It is based on the technique of *knots*, which are schematic trees of depth $\leq$1 that occur in the forest-shaped models of a knowledge base. They have been introduced in the context of non-monotonic logic programming for the class $\mathbb{FDNC}$ of programs, which have forest-shaped models [16].

The main result presented in this paper is an algorithm for answering CQs over $\mathcal{SH}$ knowledge bases. It extends a similar algorithm for $\mathcal{ALCH}$ presented in [13] and works

⋆ This work was partially supported by the FWF grant P17212, the EC NoE REWERSE (IST-506779), and the Mexican National Council for Science and Technology (CONACYT).

in exponential time for $\mathcal{ALCH}$, as well as for large classes of queries in $\mathcal{SH}$, showing that the problem is not more expensive than satisfiability testing. The algorithm is worst-case optimal and improves over previous ones that require double exponential time. Furthermore, it reconfirms Lutz's finding [9] that inverse roles make CQ answering beyond $\mathcal{ALC}$ significantly harder. Our query answering algorithm, which was developed independently from [10], has the following features:

- For a fixed terminological component and query, it can be non-deterministically run in polynomial time. Its CONP data complexity is worst-case optimal, as answering CQs is known to be CONP-complete for a wide range of DLs including $\mathcal{ALC}$; see e.g., [12].
- It provides a modular *knowledge compilation* of the TBox and the query, such that further queries can reuse the TBox compilation. In particular, queries of bounded size can be incorporated in time polynomial in the size of the compilation. This is specially useful for evaluating many queries over the same knowledge base.
- The compiled knowledge can be expressed as a (unstratified) datalog program, which is evaluated over an input set of facts (ABox) and computes the answers also for non-ground queries. This may make the algorithm more amenable for efficient implementation in practice than some of the previous automata- or tableaux-based approaches.

While we focus on $\mathcal{SH}$, the method extends to richer DLs. Indeed, once we obtain the *knot representation* of a terminology, the algorithm works on the knots and does not depend much on the constructs of the logic. Hence, in other logics supporting a knot representation, CQs may also be not more expensive than consistency. The technique opens an interesting perspective that might be exploited for other purposes as well.

2 Preliminaries

We assume countably infinite sets $\mathbf{C}$, $\mathbf{R}$ and $\mathbf{I}$ of *concept names*, *roles*, and *individuals* respectively. Further, $\mathbf{C}$ contains $\top$ and $\bot$. *Concepts (in $\mathcal{SH}$)* are inductively defined as follows: (a) every concept name $A \in \mathbf{C}$ is a concept, and (b) if C, D are concepts and $R \in \mathbf{R}$ is a role, then $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$, $\exists R.C$ are concepts.

Let C, D be concepts, R, S be roles, a, b be individuals, and A be a concept name. An expression $C \sqsubseteq D$ is a *general concept inclusion axiom (GCI)*, an expression $R \sqsubseteq S$ is a *role inclusion axiom (RI)*, an expression *Trans*(R) is a *transitivity axiom*, while expressions $a{:}A$ and $\langle a, b\rangle{:}R$ are *assertions*. An $\mathcal{SH}$ *knowledge base* (KB) is a tuple $\mathcal{K}=\langle \mathcal{T}, \mathcal{A}\rangle$, where the *TBox* $\mathcal{T}$ is a finite set of GCIs, RIs and transitivity axioms and the *ABox* $\mathcal{A}$ is a finite set of assertions. W.l.o.g. we assume $\mathcal{A} \neq \emptyset$, and that all concepts and roles occurring in $\mathcal{A}$ occur in $\mathcal{T}$. Let $\mathbf{C}(\mathcal{K})$, $\mathbf{R}(\mathcal{K})$ and $\mathbf{I}(\mathcal{K})$ respectively denote the sets of concept names, roles and individuals occurring in $\mathcal{K}$. Let $\mathbf{R}^+(\mathcal{K})=\{R \in \mathbf{R} | \mathit{Trans}(R) \in \mathcal{T}\}$, and $\sqsubseteq^*_\mathcal{T}$ be the reflexive transitive closure of $S \sqsubseteq R \in \mathcal{T}$.

We assume the reader is familiar with the standard semantics of $\mathcal{SH}$ (see, e.g., [17]). In the following, we use $\mathcal{I}$ to denote an interpretation for a KB, $\Delta^\mathcal{I}$ for the domain of $\mathcal{I}$, and $C^\mathcal{I}$ and $R^\mathcal{I}$ for the interpretation of a concept C and role R respectively.

Conjunctive Query Answering. Let $\mathbf{V}$ be a countably infinite set of variables. A *conjunctive query* (CQ, or *query*) over a KB $\mathcal{K}$ is a finite set of atoms of the form $A(x)$ or $R(x, y)$, where $A \in \mathbf{C}(\mathcal{K})$, $R \in \mathbf{R}(\mathcal{K})$ and $x, y \in \mathbf{V}$.[1] A query q is associated with a

[1] Note that no individuals occur in q. This is no limitation, as for any constant a we can use a new concept name C_a, replace a in q by a new variable y, and add $C_a(y)$ to q and $a : C_a$ to $\mathcal{A}$.

unique (possibly empty) tuple $\boldsymbol{x}$ of *answer* variables occurring in the atoms of q. By $\mathbf{V}(q)$ we denote the variables occurring in the atoms of q.

A *match for q in an interpretation $\mathcal{I}$ for $\mathcal{K}$* is a mapping θ from $\mathbf{V}(q)$ to $\Delta^{\mathcal{I}}$ s.t. (i) $\theta(x) \in A^{\mathcal{I}}$ for each $A(x) \in q$, and (ii) $\langle\theta(x), \theta(y)\rangle \in R^{\mathcal{I}}$ for each $R(x, y) \in q$. A tuple $\boldsymbol{c}$ individuals from $\mathbf{I}(\mathcal{K})$ (of the same arity as $\boldsymbol{x}$) *is an answer of q over $\mathcal{I}$*, if $\boldsymbol{c} = \theta(\boldsymbol{x})$ for some match θ for q in $\mathcal{I}$; $\mathsf{ans}(q, \mathcal{I})$ denotes all answers of q over $\mathcal{I}$. The *answer* of q over $\mathcal{K}$ is the set $\mathsf{ans}(q, \mathcal{K})$ of all tuples $\boldsymbol{c}$ s.t. $\boldsymbol{c} \in \mathsf{ans}(q, \mathcal{I})$ for every model $\mathcal{I}$ of $\mathcal{K}$.

Eliminating Transitive Roles. Each $\mathcal{SH}$ KB $\mathcal{K}$ can be rewritten in linear time into an $\mathcal{ALCH}$ KB $\mathcal{K}'$ s.t. each model of $\mathcal{K}$ is a model of $\mathcal{K}'$, and each model of $\mathcal{K}'$ can be extended to a model of $\mathcal{K}$. This can be done by deleting the transitivity axioms of $\mathcal{K}$ and adding news GCIs [18]. A CQ over $\mathcal{K}$ can then be answered using the models of $\mathcal{K}'$.

Definition 1. *Let $\mathcal{K}$ be an $\mathcal{ALCH}$ KB, q be a CQ with answer variables $\boldsymbol{x}$, and $T \subseteq \mathbf{R}(\mathcal{K})$ be a set of roles. Then a T-*match for q in an interpretation $\mathcal{I}$ for $\mathcal{K}$ *is a mapping σ from $\mathbf{V}(q)$ to $\Delta^{\mathcal{I}}$ s.t. (i) if $A(x) \in q$, then $\sigma(x) \in A^{\mathcal{I}}$; and (ii) if $R(x, y) \in q$, then $\langle\sigma(x), \sigma(y)\rangle \in R^{\mathcal{I}^{\oplus}}$, where $\mathcal{I}^{\oplus}$ is the minimal extension of $\mathcal{I}$ s.t. $R^{\mathcal{I}^{\oplus}}$ is transitively closed for every $R \in T$, and ${S_1}^{\mathcal{I}^{\oplus}} \subseteq {S_2}^{\mathcal{I}^{\oplus}}$ for every $S_1 \sqsubseteq S_2$ in $\mathcal{K}$. By $\mathsf{ans}_T(q, \mathcal{I})$ we denote the set of all tuples $\boldsymbol{c}$ of individuals s.t. $\boldsymbol{c}=\sigma(\boldsymbol{x})$ for some T-match σ for q in $\mathcal{I}$. Further, $\mathsf{ans}_T(q, \mathcal{K})$ denotes all tuples $\boldsymbol{c}$ s.t. $\boldsymbol{c} \in \mathsf{ans}_T(q, \mathcal{I})$ for each model $\mathcal{I}$ of $\mathcal{K}$.*

Theorem 1. *For any $\mathcal{SH}$ KB $\mathcal{K}$ and CQ q, we can obtain in linear time an $\mathcal{ALCH}$ KB $\mathcal{K}'$ such that $\mathsf{ans}(q, \mathcal{K}) = \mathsf{ans}_T(q, \mathcal{K}')$, where $T = \mathbf{R}^+(\mathcal{K})$.*

The theorem above follows from [18]. In the rest, we concentrate on $\mathcal{ALCH}$, and show how to compute $\mathsf{ans}_T(q, \mathcal{K})$ for a given $\mathcal{ALCH}$ KB $\mathcal{K}$, a CQ q and a set $T \subseteq \mathbf{R}(\mathcal{K})$.

3 Normal Knowledge Bases

We focus on *normal* KBs and on a restricted class models: the minimal Herbrand models of the skolemized first-order theory obtained by applying the standard translation.

Definition 2. *An $\mathcal{ALCH}$ KB $\mathcal{K}=\langle\mathcal{T}, \mathcal{A}\rangle$ is* normal *if all the GCIs in $\mathcal{T}$ are of the form (E) $A_0 \sqsubseteq \exists R.B_0$, (U) $A_0 \sqsubseteq \forall R.B_0$, or (D) $A_0 \sqcap \ldots \sqcap A_n \sqsubseteq B_0 \sqcup \ldots \sqcup B_m$, where each $A_i, B_j \in \mathbf{C}$, $n, m > 0$, and $\top$ does not occur in $\mathcal{K}$.*

For a normal KB $\mathcal{K}$, its Herbrand universe *$\mathcal{U}_{\mathcal{K}}$ is the set of all* terms *inductively defined as follows: (i) each $c \in \mathbf{I}(\mathcal{K})$ is a term, and (ii) if t is a term and α is a GCI of type (E) occurring in $\mathcal{K}$, then $f_\alpha(t)$ is a term. Let $\mathcal{B}_{\mathcal{K}}$ be the set of all* atoms *of the form $C(s)$ and $R(s, t)$ with $C \in \mathbf{C}(\mathcal{K})$, $R \in \mathbf{R}(\mathcal{K})$, and $s, t \in \mathcal{U}_{\mathcal{K}}$. An* Herbrand interpretation *of $\mathcal{K}$ is any set $I \subseteq \mathcal{B}_{\mathcal{K}}$; it represents the interpretation $\mathcal{I}$ with $\Delta^{\mathcal{I}}=\mathcal{U}_{\mathcal{K}}$, $C^{\mathcal{I}}=\{d \mid C(d) \in I\}$, $R^{\mathcal{I}}=\{\langle c, d\rangle \mid R(c, d) \in I\}$ and $c^{\mathcal{I}}=c$ for each $C \in \mathbf{C}(\mathcal{K})$, $R \in \mathbf{R}(\mathcal{K})$ and $c \in \mathbf{I}(\mathcal{K})$.*

Such an I is an S-Herbrand model *of $\mathcal{K}$, if it is a model of $\mathcal{K}$, and for each $\alpha = A \sqsubseteq \exists R.B$ in $\mathcal{K}$, $A(t) \in I$ implies $R(t, f_\alpha(t)) \in I$ and $B(f_\alpha(t)) \in I$. Moreover, I is a* minimal S-Herbrand *model of $\mathcal{K}$, if no $J \subset I$ is an S-Herbrand model of $\mathcal{K}$. We denote by $\mathcal{M}(\mathcal{K})$ the set of all minimal S-Herbrand models of $\mathcal{K}$.*

Using well-known structural transformations, every $\mathcal{K}$ can be rewritten into a normal $\mathcal{K}'$ while preserving the query answers. Further, one can show via the first-order logic that $\mathcal{M}(\mathcal{K})$ suffices to answer a CQ over $\mathcal{K}$. In the following, by 'interpretation' (resp., 'minimal model') we mean Herbrand interpretation (resp., minimal S-Herbrand model).

Theorem 2. *Given an $\mathcal{ALCH}$ KB $\mathcal{K}$, a CQ q and a set $T \subseteq \mathbf{R}(\mathcal{K})$, we can obtain in linear time a normal KB $\mathcal{K}'$ such that $\mathsf{ans}_T(q, \mathcal{K}) = \bigcap_{I \in \mathcal{M}(\mathcal{K}')} \mathsf{ans}_T(q, I)$.*

4 Knots

In what follows, let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be an arbitrary normal $\mathcal{ALCH}$ KB. We provide a method for finitely representing the possibly infinite minimal models of $\mathcal{K}$; it exploits the *forest-shaped model property* which allows us to view each minimal model of $\mathcal{K}$ as a graph and a set of trees rooted at nodes of the graph. A set of atoms is *forest-shaped* if its binary atoms are of the form $R(a, b)$ or $R(t, f(t))$ for a term t and individuals a, b.

Proposition 1. *Every $I \in \mathcal{M}(\mathcal{K})$ is forest-shaped.*

Due to the above, minimal models of $\mathcal{K}$ can be composed out of trees of depth ≤ 1 that we call *knots*. We write $t \hat{\in} I$ if a set of atoms I contains an atom with the term t as argument, and denote by $\mathcal{U}(I)$ the set of all terms $t \hat{\in} I$.

Definition 3. *(Knots) A* knot with root (term) t *is a set of atoms K such that each atom in K is of the form $A(t)$, $R(t, f(t))$, or $A(f(t))$ where A, R, and f are arbitrary;* $\mathsf{succ}(K)$ *denotes the set of terms of the form $f(t) \hat{\in} K$.*

A knot with root term t can be viewed as a labeled tree of depth at most 1, whose nodes and edges are labeled with concept names and roles respectively. In the following we only consider knots whose concept names are from $\mathbf{C}(\mathcal{K})$ and whose roles from $\mathbf{R}(\mathcal{K})$ (note that no restriction is imposed on t). For a term t, let $\mathcal{B}_t$ denote the set of all atoms that can be built from $\mathbf{C}(\mathcal{K})$ and $\mathbf{R}(\mathcal{K})$ using t and terms of the form $f(t)$ as arguments. Note that for a forest-shaped interpretation I for $\mathcal{K}$ and $t \hat{\in} I$, the set $I \cap \mathcal{B}_t$ is a knot, and that $\emptyset$ is a knot with an arbitrary root. We introduce *min-consistent* knots, which are self-contained model building blocks for minimal models of $\mathcal{K}$.

Definition 4. *Given a knot K with root t, we say K is* consistent *(w.r.t. $\mathcal{K}$), if:*
(a) $\bot(u) \notin K$ *for each* $u \in \{t\} \cup \mathsf{succ}(K)$.
(b) if $A \sqsubseteq \forall R.B \in \mathcal{T}$, $A(t) \in K$ *and* $R(t, f(t)) \in K$, *then* $B(f(t)) \in K$;
(c) if $\alpha = A \sqsubseteq \exists R.B$, $\alpha \in \mathcal{T}$ *and* $A(t) \in K$, *then* $R(t, f_\alpha(t)) \in K$ *and* $B(f_\alpha(t)) \in K$;
(d) if $A_0 \sqcap \ldots \sqcap A_n \sqsubseteq B_0 \sqcup \ldots \sqcup B_m \in \mathcal{T}$, $s \in \mathsf{succ}(K)$ *and* $\{A_0(s), \ldots, A_n(s)\} \subseteq K$, *then* $B_i(s) \in K$ *for some* B_i;
(e) if $R \sqsubseteq S \in \mathcal{T}$ *and* $R(t, f(t)) \in K$, *then* $S(t, f(t)) \in K$.
K *is* min-consistent *if each $K' \subset K$ obtained from K by removing atoms where an $s \in \mathsf{succ}(K)$ occurs is inconsistent.*

Intuitively, given a term t and a set of concepts it satisfies, a min-consistent knot with root t encodes a possible combination of immediate successors for t in a model of $\mathcal{K}$. The tree-parts of the forest shaped models of $\mathcal{K}$ will be represented by min-consistent knots. Now we introduce some notions for dealing with the graph part.

Definition 5. *The KB $\mathcal{K}^g$ is obtained from $\mathcal{K}$ by deleting all axioms of type (E) as in Definition 2. A set of atoms G is a* min-graph *of $\mathcal{K}$ if $G \in \mathcal{M}(\mathcal{K}^g)$.*

The minimal models of $\mathcal{K}$ can be characterized in terms of min-graphs and min-consistent knots. For a set of atoms I, let I^g contain all atoms $A(a)$, $R(a, b)$ in I with $a, b \in \mathbf{I}(\mathcal{K})$.

Theorem 3. *If I is an interpretation for $\mathcal{K}$, then $I \in \mathcal{M}(\mathcal{K})$ iff I is forest-shaped, I^g is a min-graph of $\mathcal{K}$, and for each term $t \hat{\in} I$, the knot $I \cap \mathcal{B}_t$ is min-consistent w.r.t. $\mathcal{K}$.*

Due to the above theorem, one can view each minimal model of $\mathcal{K}$ as being constructed out of a min-graph and a set of min-consistent knots. The set of knots may be infinite, but only finitely many of them are non-isomorphic modulo the root term.

Following the observation above, we represent all minimal models using a finite set of knots. Let $\mathbf{x}$ be an individual not occurring in any $\mathcal{ALCH}$ KB. We say a knot K with root t is *abstract*, if $t = \mathbf{x}$. A knot K' with root u is an *instance* of an abstract knot K, if K' can be obtained from K by replacing each occurrence of $\mathbf{x}$ with u. Given a set of knots L, we define the conditions that ensure that we can construct tree-shaped parts of minimal models using the knots in L. Intuitively, for each knot K in L and for each $s \in \mathsf{succ}(K)$, there must be some knot that can be instantiated at s. This ensures that trees where all the nodes have the necessary successors can be built. Given two sets of atoms I and J, we write $I_t \approx J_u$ if $\{A \mid A(t) \in I\} = \{A \mid A(u) \in J\}$.

Definition 6. *Let L be a set of abstract knots. Given $K \in L$ and $s \in \mathsf{succ}(K)$, we say $K' \in L$* is an s-successor of K *if $K_s \approx K'_{\mathbf{x}}$; the set of s-successors of K in L is denoted $L[K, s]$. If every $K \in L$ is min-consistent w.r.t. $\mathcal{K}$ and $L[K, s] \neq \emptyset$ for each $K \in L$ and each $s \in \mathsf{succ}(K)$, then L is* $\mathcal{K}$-founded.

We show how minimal models of $\mathcal{K}$ can be constructed from a $\mathcal{K}$-founded set of knots L, and describe a set L that generates all the models in $\mathcal{M}(\mathcal{K})$.

Definition 7. *We say that I* is generated *by a $\mathcal{K}$-founded knot set L if I is a $\subseteq$-minimal interpretation containing some min-graph G of $\mathcal{K}$ and, for every term $t \hat{\in} I$, $I \cap \mathcal{B}_t$ is an instance of some $K \in L$. The set of interpretations generated by L is denoted $\mathcal{F}_{\mathcal{K}}(L)$.*

The set $\mathcal{F}_{\mathcal{K}}(L)$ represents all the forest-shaped interpretations that can be built from a min-graph by instantiating the knots in L. Importantly, such interpretations are actually minimal models; due to Theorem 3, if L is $\mathcal{K}$-founded and $I \in \mathcal{F}_{\mathcal{K}}(L)$, then $I \in \mathcal{M}(\mathcal{K})$.

Definition 8. *The smallest set of abstract knots that contains every $\mathcal{K}$-founded set of knots is denoted $\mathbb{K}_{\mathcal{K}}$.*

The crucial property of $\mathbb{K}_{\mathcal{K}}$ is that it captures the tree-structures of the minimal models of $\mathcal{K}$, and together with the min-graphs, it captures all the minimal models of $\mathcal{K}$.

Theorem 4. *$\mathbb{K}_{\mathcal{K}}$ is $\mathcal{K}$-founded, and $\mathcal{F}_{\mathcal{K}}(\mathbb{K}_{\mathcal{K}}) = \mathcal{M}(\mathcal{K})$.*

5 Query Answering with Knots

In what follows, we assume an $\mathcal{ALCH}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, a $\mathcal{K}$-founded set of knots L, a $T \subseteq \mathbf{R}(\mathcal{K})$, and a CQ q. For the sake of this paper, we assume that if R occurs in q and $R' \sqsubseteq^*_{\mathcal{T}} R$ for some $R' \in T$, then $R \in T$; such an R is called *T-safe*.[2] We present a method for computing $\mathsf{ans}_T(q, I)$ for each $I \in \mathcal{F}_{\mathcal{K}}(L)$. By Theorem 4, setting $L = \mathbb{K}_{\mathcal{K}}$ allows us to compute $\mathsf{ans}_T(q, \mathcal{K})$ and, by Theorems 1 and 2, to answer CQs over $\mathcal{SH}$.

To develop our query answering algorithm, we first define the entailment of *subqueries at a knot K*, which informally means that there is a match for some parts of the query in each *tree* that is generated from L and starts with K, and provide a decision procedure for it. The method is based on a fixpoint computation that derives in each iteration new pairs of knots and subqueries for which the entailment relation holds, based on previously computed pairs. To decide whether a given a knot K entails a subquery, the subqueries that the possible successors knots of K entail are considered. Hence, the algorithm "back-propagates" the information via the possible successor relation.

[2] Note that this imposes no restrictions for query answering in $\mathcal{ALCH}$ or $\mathcal{S}$.

In a second stage, we consider the min-graphs of $\mathcal{K}$ and verify whether for each min-graph G, the query can be mapped in each forest-shaped minimal model of $\mathcal{K}$ that is built from G and the knots in $\mathbb{K}_{\mathcal{K}}$. To this end, we verify if, for any possible way of constructing a model out of G, a mapping for the full query can be composed from some partial mapping of q into G and some mappings that exist in the trees rooted at the individuals. The existence of the latter mappings will be witnessed by the precomputed set of pairs of knots and subqueries for which the entailment relation holds.

Since the minimal models of $\mathcal{K}$ are forest-shaped, for any query match π and any tree shaped part I of a model, the image of the subquery of q that is mapped inside I under π is a subtree of I. This implies, e.g., that if two atoms $R(x,y)$, $R'(x',y)$ of q are mapped inside a tree, then x and x' must be mapped on the same path. Moreover, if R and R' are not in T (i.e., they are not transitive), then x and x' must be mapped to the same node. In general, each x induces a set of variables V^T_x which are mapped into the subtree rooted at the map of x if the latter is inside a tree. This set contains x, the successors of x, and each variable that must be mapped to the same node as one of them because they have a common non-transitive successor. For other variables y in $\mathbf{V}(q)$, x may not determine whether y is mapped above or below it. This is the case, e.g., if y is neither a predecessor nor a successor of x and they have a common transitive successor. However, if we fix a set X of such nodes, they will induce a unique set of variables V^T_X that are mapped below them in any a query match mapping X into a tree.

Definition 9. *Assume a variable $x \in \mathbf{V}(q)$ and a set $X \subseteq \mathbf{V}(q)$. Let $\mathsf{R}_0(x) = \{x\}$ and $\mathsf{R}_{n+1}(x) = \{y \in \mathbf{V}(q) \mid R(x',y) \in q$ and $x' \in \mathsf{R}_n(x)\}$, for every $n \geq 0$. We also define next(x)=$\mathsf{R}_1(x)$ and prev(x) $= \{y \in \mathbf{V}(q) \mid x \in$ next(y)$\}$. By V^T_x we denote the smallest subset of $\mathbf{V}(q)$ s.t.* (i) *$\mathsf{R}_n(x) \in \mathsf{V}^T_x$ for every $n \in \omega$, and* (ii) *$R(y,z) \in q$, $R(y',z) \in q$, $\{R,R'\} \cap T = \emptyset$ and $y \in \mathsf{V}^T_x$ imply $y' \in \mathsf{V}^T_x$. By V^T_X we denote $\bigcup_{x \in X} \mathsf{V}^T_x$.*

Definition 10. *A* canonical rooting set *of q is a $\subseteq$-maximal set V of sets of variables $X \subseteq \mathbf{V}(q)$ such that $\mathsf{V}^T_X \neq \mathsf{V}^T_Y$ for every $X, Y \in V$.*

In what follows, we assume a fixed arbitrary canonical rooting set $\mathbb{V}^T_q$ of q. We are ready to formally define the subqueries and their entailment in trees.

Definition 11. *For a knot $K \in L$, I* is a tree generated by *L (starting with K), if I is a $\subseteq$-minimal set of atoms such that $K \subseteq I$ and, for each term $t \hat{\in} I$, $I \cap \mathcal{B}_t$ is an instance of some $K' \in L$. We denote by $\mathcal{T}(L,K)$ the set of all such trees.*

For a set of atoms I, let I^T be its minimal extension s.t. (i) *if $\{R(a,b), R(b,c)\} \subseteq I^T$ and $R \in T$, then $R(a,c) \in I^T$; and* (ii) *if $R(a,b) \in I^T$ and $R \sqsubseteq S \in \mathcal{T}$, then $S(a,b) \in I^T$.*

Definition 12. *A* disjunctive subquery *of q is a set $\rho_q \subseteq \mathbb{V}^T_q$. By $\mathbb{R}_q$ we denote the set of all disjunctive subqueries of q. For a tree I generated by L, a* rooted match for *$X \in \rho_q$* in *I is a function π from V_X to $\mathcal{U}(I)$ s.t. for each $x, y \in X$:*
(RM1) if $A(x) \in q$ then $A(\pi(x)) \in I$;
(RM2) if $R(x,y) \in q$ then $R(\pi(x), \pi(y)) \in I^T$;
(RM3) if $y \in \mathsf{V}_X$, $R(x,y) \in q$ and $x \notin \mathsf{V}_X$, then $\pi(y) = \mathbf{x}$, or $R \in T$ and $R(\mathbf{x}, \pi(y)) \in I^T$.
We write $I \models \rho_q$ if for some $X \in \rho_q$ there exists a rooted match in I. Further, $I \models^d X$ holds if for some $X \in \rho_q$ there is a rooted match π in I s.t. for each $y \in \mathsf{V}_X$, the depth of the term $\pi(y)$ is $\leq d$. We write $K \models_L \rho_q$ (resp., $K \models^d_L \rho_q$) if for each $I \in \mathcal{T}(L,K)$ we have $I \models \rho_q$ (resp., $I \models^d \rho_q$). We omit the subscripts if clear from the context.

Note that the trees in $\mathcal{T}(L,K)$ have root $\mathbf{x}$. Intuitively, a rooted match for X in a tree I is a homomorphic embedding of the subquery of q induced by V_X into I. Further, each $y \in \mathsf{V}_X$ that has some predecessor variable not in V_X must be mapped to $\mathbf{x}$ or reachable from it via a path suitably labeled with a transitive role (RM3). A rooted match for X can be part of a full query match in a model containing an instance of I, provided that all the predecessors of y have a match in it which is above the (sub)tree instantiating I.

We construct a set $\Gamma(L,q)$ of all pairs (K,ρ) such that $K \models_L \rho$. We first compute the pairs (K,ρ) with $K \models_L^0 \rho$, and then continue via fixpoint iteration to obtain the pairs (K,ρ) with $K \models_L^d \rho$ for an arbitrary $d \in \omega$. Such pairs capture the $\models_L$ relation:

Proposition 2. *If $K \models_L \rho$, then there exists $d \in \omega$ such that $K \models_L^d \rho$.*

A key part of the algorithm is to characterize the minimal sets $l \subseteq \mathbb{V}_q^L$ such that there is a tree starting at K that models exactly the sets in l. To this aim, we employ *minimal hitting sets*. Informally, we can see these sets as the most general way of 'grouping' the trees by the exact elements of $\mathbb{V}_q^L$ for which they provide a match of bounded depth.

Definition 13. *Assume a knot $K \in L$ and a set $S \subseteq L \times \mathbb{R}_q$. A set $l \subseteq \mathbb{V}_q^T$ is a* minimal hitting set of S w.r.t. K *if it is a $\subseteq$-minimal set s.t. $l \cap \rho \neq \emptyset$ for every $(K,\rho) \in S$.*

Proposition 3. *Assume $K \in L$, $d \in \omega$ and let S be the set of all pairs (K,ρ), $\rho \in \mathbb{R}_q$, such that $K \models_L^d \rho$. If l is a minimal hitting set of S w.r.t. K, then there is some $I \in \mathcal{T}(L,K)$ such that, for every $X \in \mathbb{V}_q^T$, $I \models^d \{X\}$ iff $X \in l$.*

We now sketch the procedure for computing $\models_L$. For each $d \in \omega$, let S^d denote the set of all pairs (K,ρ) s.t. $K \models_L^d \rho$. As easily seen, the set S^0 can be computed by checking which sets in $\mathbb{V}_q^L$ can be satisfied by direct mappings into the roots of the knots in L.

For the inductive case, suppose for some $d \in \omega$ we have computed the set S^d. Assume some ρ and an arbitrary knot $K \in L$. To verify whether $K \models_L^{d+1} \rho$, we consider K*-hits* which capture the possible ways of choosing for each $s \in \mathsf{succ}(K)$ a knot $K' \in L[K,s]$ and a minimal hitting set l of S^d w.r.t. K'. Intuitively, we conclude $K \models_L^{d+1} \rho$ if for each K-hit there is an $X \in \rho$ such that part of V_X can be mapped into K, while the rest of the variables are contained in the chosen minimal hitting sets; this partitioning of V_X will be captured by the notion of K*-mapping*.

Definition 14. *A* successor choice *for $K \in L$ is a function mapping each $s \in \mathsf{succ}(K)$ to a $K' \in L[K,s]$. A* K-hit of $S \subseteq L \times \mathbb{R}_q$ *is a pair (sc, hs) of a successor choice sc for K and a function hs mapping each $s \in \mathsf{succ}(K)$ to a minimal hitting set of S w.r.t. $sc(s)$.*

Now we introduce K-mappings which are composed of two sets r and o of variables, and a function $b(\cdot)$ that maps variables to leaves of K. The variables in r have a match at the *root* of K, while the variables in b have a match *below* the root of K. The variables in o are predecessors of variables in b and don't have a mapping in the trees rooted at K: an x in o simply indicates that there is a transitive path leading to its successor in b. Intuitively, in order for a K-mapping to represent a rooted match in a tree starting with K, each x in the domain of b must have a match in the subtree with root $b(x)$. In particular, the latter holds whenever each such x is in the hitting set $hs(b(x))$ of some K-hit; if this is the case, we say that the K-hit *complies* with the K-mapping. The domain of a function g from A to B is denoted $\mathsf{dom}(g)$, and $g^{-1}(b) = \{a \in A \mid g(a) = b\}$.

Definition 15. *For a knot $K \in L$, a K-*mapping *for q is a tuple $m{=}\langle r, o, b\rangle$, where $r \subseteq \mathbf{V}(q)$, $o \subseteq \mathbf{V}(q)$ and b is a partial function from $\mathbf{V}(q)$ to $\mathsf{succ}(K)$ s.t. r, o and $\mathsf{dom}(b)$ are pairwise disjoint and:*
(M1) $x \in r$ and $A(x) \in q$ imply $A(\mathbf{x}) \in K$;
(M2) $x \in r \cup \mathsf{dom}(o)$ and $R(x, y) \in q$ imply $y \in \mathsf{dom}(b)$ and $R(\mathbf{x}, b(y)) \in K$;
(M3) $x \in \mathsf{dom}(o)$ and $R(x, y) \in q$ imply $R \in T$; and
(M4) for each $s \in \mathsf{succ}(K)$, prev$(b^{-1}(s)) \subseteq r \cup o \cup b^{-1}(s)$ and next$(b^{-1}(s)) \subseteq b^{-1}(s)$.
We define roots$(m) = r \cup \{x \in \mathsf{dom}(b) \mid \textit{prev}(x) = \emptyset\}$. A K-hit (sc, hs) complies with *m if for all $s \in \mathsf{succ}(K)$ with $b^{-1}(s) \neq \emptyset$, there is some $X{\in}$hs(s) s.t. $b^{-1}(s){=}\mathsf{V}_X$.*

We are ready to define a relation $S \vdash_{L,q} (K, \rho)$ for obtaining new pairs (K, ρ) with $K \models \rho$, which follow from a given set S of pairs (K', ρ') with $K' \models \rho'$.

Definition 16. *Assume $K \in L$, $\rho \in \mathbb{R}_q$ and a set $S \subseteq L{\times}\mathbb{R}_q$. The pair (K, ρ)* follows from *S, in symbols $S \vdash_{L,q} (K, \rho)$, if for every K-hit k of S there is a K-mapping m such that k complies with m and $\mathsf{V}_{\textit{roots}(m)} = \mathsf{V}_X$ for some $X \in \rho$.*

Definition 17. *The set $\Gamma(L, q) \subseteq L \times, \mathbb{R}_q$ is defined as $\Gamma(L, q){=}\bigcup_{d \in \omega} \Gamma(L, q)^d$, where $\Gamma(L, q)^0{=}\{(K, \rho) \mid \emptyset \vdash_{L,q} (K, \rho)\}$ and $\Gamma(L, q)^{d+1}{=}\{(K, \rho) \mid \Gamma(L, q)^d \vdash_{L,q} (K, \rho)\}$.*

Note that $\Gamma(L, q)^0 \subseteq \Gamma(L, q)^1 \subseteq \cdots \subseteq \Gamma(L, q)^d$ for each $d \in \omega$. Since $L \times \mathbb{R}_q$ is finite, $\Gamma(L, q)$ is finite and unique. Furthermore, for every $d \in \omega$, $K \models^d_L \rho$ iff $(K, \rho) \in \Gamma(L, q)^d$. Hence, $\Gamma(L, q)$ captures the $\models_L$ relation. The (inductive) proof of this correspondence can be found in the extended version of this paper.

Theorem 5. *For $(K, \rho){\in} L{\times}\mathbb{R}_q$, $K{\models_L}\rho$ iff $(K, \rho){\in}\Gamma(L, q)$.*

Now that we can decide subquery entailment at the knots of L, we move to query answering over $\mathcal{K}$. By Theorems 1 and 4, it suffices to consider the forest-shaped models constructed from the min-graphs of $\mathcal{K}$ and the trees generated from $\mathbb{K}_{\mathcal{K}}$. The machinery we have presented deals with the parts of query matches that occur inside the trees, now we extend it to deal with the min-graph part. In what follows we assume that q has answer variables $\boldsymbol{x}$, while $\boldsymbol{c}$ is a tuple of individuals of the same arity as $\boldsymbol{x}$.

Definition 18. *An extended min-graph H of $\mathcal{K}$ is a $\subseteq$-minimal set of atoms containing a min-graph of $\mathcal{K}$ and s.t. for each individual a, $H \cap \mathcal{B}_a$ is an instance of a knot in $\mathbb{K}$.*

Consider a min-graph G. Intuitively, each extended min-graph containing G can be viewed as a "super" knot whose root is G, while its leaves are the leaves of the knots that extend G. Given this similarity, the full query q can be answered by adjusting the notions of K-hits and K-mappings to deal with extended graphs.

Definition 19. *Let H be an extended min-graph. A* successor choice *for H is a function that maps each term $f(c)\hat{\in}H$ to some $K \in L$ such that $H_{f(c)} \approx K_{\mathbf{x}}$. Then an H-*hit *is a pair (sc, hs), where sc is a successor choice for H, and hs is a function that maps each $f(c)\hat{\in}H$ to a minimal hitting set of $\Gamma(\mathbb{K}_{\mathcal{K}}, q)$ w.r.t. the knot sc$(f(c))$.*

We now provide a method to decide the existence of a query match in all models starting with each extended graph H. Similarly as for the knots, we consider all H-hits and check if they comply with the different partial mappings into the extended graph H.

Definition 20. *Let $q_{\downarrow V}$ be the restriction of the query q to the atoms containing variables in V. An H-hit (sc, hs)* complies with *a constant tuple $\boldsymbol{c}$ if there exist disjoint sets $V, V' \subseteq \mathbf{V}(q)$ and a function π from $V \cup V'$ to $\mathcal{U}(H)$ s.t. $\pi^{-1}(V') \cap \mathbf{I} = \emptyset$ and:*

- *for each $x \in V$, $A(x) \in q$ implies $A(\pi(x)) \in H$;*
- *for each $x, y \in V \cup V'$, $R(x, y) \in q$ implies $R(\pi(x), \pi(y)) \in H^T$;*
- *for each $f(c)$ with $\pi^{-1}(f(c)) \neq \emptyset$ there is an $X \in hs(f(c))$ s.t. $\mathsf{V}_X = \mathsf{V}_{\pi^{-1}(f(c))}$;*
- *for each $y \in \mathbf{V}(q) \backslash (V \cup V')$, there is an $f(c) \hat{\in} H$ and an $X \in hs(f(c))$ s.t. $y \in \mathsf{V}_X$;*
- *for each answer variable x_i, $\pi(x_i) = c_i$.*

Let $\mathbb{C}_{\mathcal{K}}$ be the set of all tuples (H, sc, hs) s.t. H is an extended min-graph of $\mathcal{K}$ and (sc, hs) is an H-hit. For each such tuple $\lambda = (H, sc, hs)$ in $\mathbb{C}_{\mathcal{K}}$, we define $\mathsf{ans}_T(q, \lambda)$ as the set of all tuples $\boldsymbol{c}$ that comply with the H-hit (sc, hs).

Theorem 6. *For every CQ q over $\mathcal{K}$, it holds that $\mathsf{ans}_T(\mathcal{K}, q) = \bigcap_{\lambda \in \mathbb{C}_{\mathcal{K}}} \mathsf{ans}_T(q, \lambda)$.*

The proof is similar to the inductive step of the one of Theorem 5 (see extended paper).

We remark that for a given $\lambda = (H, sc, hs)$ in $\mathbb{C}_{\mathcal{K}}$, the set $\mathsf{ans}_T(q, \lambda)$ can be computed by evaluating a union of CQs over a set of atoms H_λ obtained by augmenting H with atoms that capture the variable choice (sc, hs). Furthermore, the H_λ for all the $\lambda \in \mathbb{C}_{\mathcal{K}}$ can be generated in models of a datalog program (with unstratified negation). Hence, deriving $\mathsf{ans}_T(\mathcal{K}, q)$ is reducible to computing cautious consequence in datalog.

6 Computational Complexity

Next we analyze the complexity of our algorithm. Note that, given q and T, all canonical rooting sets have equal cardinality, and recall that $\mathbb{V}_q^T$ denotes one (fixed arbitrary) such set. Recall also that only queries all whose roles are T-safe are considered.

Theorem 7. *Given a normal $\mathcal{ALCH}$ KB $\mathcal{K}$, a CQ q, a set $T \subseteq \mathbf{R}(\mathcal{K})$ and a tuple of individuals $\boldsymbol{c}$, deciding whether $\boldsymbol{c} \in \mathsf{ans}_T(q, \mathcal{K})$ is feasible in time double exponential in $|\mathcal{K}| + |q|$. Furthermore, for all instances for which $|\mathbb{V}_q^T|$ is polynomial in $|q| + |T|$, the problem can be decided in single exponential time.*

Proof. (Sketch) Let $cs := |\mathcal{K}| + |q|$. The result follows from the following observations:

• The number of distinct abstract knots over the signature of $\mathcal{K}$ is (single) exponential in cs, and $\mathbb{K}_{\mathcal{K}}$ can be constructed in exponential time by the procedure sketched in [13].
• For a given $S \subseteq \mathbb{K}_{\mathcal{K}} \times \mathbb{R}_q$ and $(K, \rho) \in \mathbb{K}_{\mathcal{K}} \times \mathbb{R}_q$, verifying $S \vdash_{\mathbb{K}_{\mathcal{K}}, q} (K, \rho)$ is feasible in time exponential in cs. Due to the monotonicity of $\vdash_{\mathbb{K}_{\mathcal{K}}, q}$ and the fact that $|\mathbb{K}_{\mathcal{K}} \times \mathbb{R}_q|$ is double exponentially bounded in cs, the set $\Gamma(L, q)$ can be computed in time double exponential in cs. If $|\mathbb{V}_q^T|$ is polynomial in $|q| + |T|$, the set $|\mathbb{K}_{\mathcal{K}} \times \mathbb{R}_q|$ is bounded by an exponential and $\Gamma(L, q)$ can be computed in time exponential in cs.
• For a given tuple $\lambda = (H, sc, hs)$ in $\mathbb{C}_{\mathcal{K}}$, the set $\mathsf{ans}(q, \lambda)$ can be computed in time exponential in cs. Indeed, for any tuple $\boldsymbol{c}$ of constants, the compliance of $\boldsymbol{c}$ with the H-hit (sc, hs) can be decided in exponential time in cs, and there are only exponentially many tuples of constants from $\mathcal{K}$ matching the arity of the answer variables of q.
• The set $|\mathbb{C}_{\mathcal{K}}|$ is double exponential in cs, and $\boldsymbol{c} \in \mathsf{ans}(\mathcal{K}, q)$ can be verified in time double exponential in cs. If $|\mathbb{V}_q^T|$ is polynomial in $|q| + |T|$, the set $|\mathbb{C}_{\mathcal{K}}|$ is bounded by an exponential, and hence $\boldsymbol{c} \in \mathsf{ans}(\mathcal{K}, q)$ can be decided in time exponential in cs. $\square$

By Theorems 1 and 2, CQ answering over $\mathcal{SH}$ KBs is reducible to CQ answering over normal $\mathcal{ALCH}$ KBs. As consistency testing in $\mathcal{ALCH}$ is EXPTIME-hard [15], we have:

Corollary 1. *Given an $\mathcal{SH}$ KB $\mathcal{K}$, a CQ q and a tuple of individuals $\boldsymbol{c}$, deciding whether $\boldsymbol{c} \in \mathsf{ans}(q, \mathcal{K})$ is EXPTIME-complete in combined complexity, provided that $|\mathbb{V}_q^T|$ is polynomial in $|q|+|\mathbf{R}^+(\mathcal{K})|$. In particular, this holds if $\mathcal{K}$ has no transitivity axioms, as $|\mathbb{V}_q^T| \leq |\mathbf{V}(q)|$. Thus CQ answering over $\mathcal{ALCH}$ KBs is EXPTIME-complete in combined complexity.*

Note that computing $\mathsf{ans}(\mathcal{K}, q)$ is also exponential in the size of the input.

We provide a syntactic restriction to obtain classes of CQs for which the sets $\mathbb{V}_q^T$ are of polynomial size in $|q| + |T|$ and hence allow for query answering in EXPTIME.

Definition 21. *Let $\mathsf{R}_+(x) = \bigcup_{i \geq 1} \mathsf{R}_i(x)$. Then a variable $x \in \mathbf{V}(q)$ is* an ABox variable, *if there is some $y \in \mathbf{V}(q)$ such that $x, y \in \mathsf{R}_+(y)$, i.e., x reaches some cycle.*

A set of variables $X \subseteq \mathbf{V}(q)$ is called connected, *if the query graph induced by q (with nodes X and an edge (x, y) iff some atom $R(x, y) \in q$ exists) is connected. The* order-freeness degree of X, *denoted $\mathsf{ofd}_q(X)$, is the size of the largest subset $X' \subseteq X$ s.t. for each $x \neq y \in X$, it holds that $y \notin \mathsf{R}_+(x)$ and $x \notin \mathsf{R}_+(y)$. The* order-freeness degree of q, *denoted $\mathsf{ofd}(q)$, is the maximum $\mathsf{ofd}_q(X)$ over all connected $X \subseteq \mathbf{V}(q)$.*

Proposition 4. *For every conjunctive query q such that $\mathsf{ofd}(q)$ is bounded by a constant, $|\mathbb{V}_q^T|$ is polynomial in $|q| + |T|$ for any T.*

The above implies that answering CQs with bounded order-freeness degree over $\mathcal{SH}$ KBs is EXPTIME-complete. Unfortunately, the order-freeness degree is often unbounded, even for simple queries (e.g., for some tree shaped queries). To address this we introduce *frame queries*, which capture more precisely the structural complexity of CQs.

Definition 22. *For a CQ q and a set $T \subseteq \mathbf{R}$, the* frame query q^T *is obtained from q by*
(1) removing all atoms where only ABox variables occur;
(2) applying each of the following rules exhaustively to the query resulting from (1):
(2.a) if there are two atoms $R(x, y)$ and $R'(x', y)$ s.t. $R, R' \notin T$, replace x with x';
(2.b) remove all atoms $R(x, y)$ s.t. next$(y)=\emptyset$ and $|$prev$(y)|=1$.

Proposition 5. *For each conjunctive query q and each set $T \subseteq \mathbf{R}$ of roles such that $\mathsf{ofd}(q^T)$ is bounded by a constant, the size of $\mathbb{V}_q^T$ is polynomial in $|q| + |T|$.*

As a result, for an $\mathcal{SH}$ KB $\mathcal{K}$ with $T = \mathbf{R}^+(\mathcal{K})$ and any CQ q such that $\mathsf{ofd}(q^T)$ is bounded by a constant, CQ answering is feasible in exponential time. Note that $\mathsf{ofd}(q^T)$ is bounded by a constant if constantly many variables occur in atoms with roles from T.

Finally, we remark that the algorithm can be easily adjusted to run in CONP in the size of $\mathcal{A}$, i.e., in data complexity; see [13] and the extended version of this paper.

7 Conclusion

We presented a novel algorithm for conjunctive query answering over DL knowledge bases, which is based on the concept of knots that have been originally conceived in the context of logic programming. It employs a technique that is different from previous query answering techniques, yet not completely unrelated; for space reasons, a respective comparison is relegated to an extended paper.

We confine here to the resolution-based method by Hustadt et al. [6], which is perhaps most closely related to ours. Similar as in our approach, their method first “compiles” the knowledge base and the query into a special form, and then exploits the possibility to answer the query by means of a datalog program. However, this is done on different grounds: the knot technique is model-theoretic in nature, while Hustadt et al.’s method is proof-theoretic, cleverly exploiting resolution and superposition machinery.

The method we presented is extendible to richer DLs beyond $\mathcal{SH}$; e.g., number restrictions can be accommodated by suitably adapting the knot representation of knowledge bases. Future work will consider such extensions, as well as more expressive queries. We believe that the knot technique can be useful in this and in other contexts.

References

1. Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2006. Data complexity of query answering in description logics. In *Proc. KR’06*, 260–270. AAAI Press.
2. Calvanese, D.; De Giacomo, G.; and Lenzerini, M. 1998. On the decidability of query containment under constraints. In *Proc. ACM PODS’98*, 149–158.
3. Calvanese, D.; Eiter, T.; and Ortiz, M. 2007. Answering regular path queries in expressive description logics: An automata-theoretic approach. In *Proc. AAAI’07*, 391–396.
4. Glimm, B.; Horrocks, I.; Lutz, C.; and Sattler, U. 2007. Conjunctive query answering for the description logic $\mathcal{SHIQ}$. In *Proc. IJCAI’07*, 399–404.
5. Glimm, B.; Horrocks, I.; Sattler, U.; 2007. Conjunctive query entailment for $\mathcal{SHOQ}$. In *Proc. DL’07*, 65–75.
6. Hustadt, U.; Motik, B.; and Sattler, U. 2005. Data complexity of reasoning in very expressive description logics. In *Proc. IJCAI’05*, 466–471.
7. Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2007. Conjunctive queries for a tractable fragment of OWL 1.1. In *Proc. ISWC/ASWC’07*, LNCS 4825, 310–323. Springer.
8. Levy, A. Y., and Rousset, M.-C. 1998. Combining Horn rules and description logics in CARIN. *Artificial Intelligence* 104(1–2):165–209.
9. Lutz, C. 2007. Inverse roles make conjunctive queries hard. In *Proc. DL’07*, 100–111.
10. Lutz, C. 2008. Two Upper Bounds for Conjunctive Query Answering in $\mathcal{SHIQ}$. In *Proc. DL’08*, to appear.
11. Ortiz, M.; Calvanese, D.; and Eiter, T. 2006. Characterizing data complexity for conjunctive query answering in expressive description logics. In *Proc. AAAI’06*, 275–280.
12. Ortiz, M.; Calvanese, D.; and Eiter, T. 2008. Data Complexity of Query Answering in Expressive Description Logics via Tableaux. Journal of Automated Reasoning, to appear.
13. Ortiz, M.; Šimkus, M.; and Eiter, T. 2008. Conjunctive Query Answering for an Expressive Description Logic without Inverses. In *Proc. AAAI’08*, to appear.
14. Rosati, R. 2007. On conjunctive query answering in $\mathcal{EL}$. In *Proc. DL’07*, 451–458.
15. Schild, K. 1991. A Correspondence Theory for Terminological Logics: Preliminary Report. In *Proc. IJCAI’91*.
16. Šimkus, M. and Eiter, T. 2007. FDNC: Decidable non-monotonic disjunctive logic programs with function symbols. In *Proc. LPAR’07*, LNCS 4790, 514–530. Full paper INFSYS RR-1843-08-01. `http://www.kr.tuwien.ac.at/research/reports/rr0801.pdf`
17. Tessaris, S. 2001. *Questions and Answers: Reasoning and Querying in Description Logic*. Ph.D. Dissertation, Univ. Manchester, CS Dept.
18. Tobies, S. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 2001.