Finding Good Proofs for Answers to Conjunctive Queries Mediated by Lightweight Ontologies

Christian Alrabbaa, Stefan Borgwardt, Patrick Koopmann and Alisa Kovtunova

Institute of Theoretical Computer Science, Technische Universität Dresden, 01062 Dresden, Germany

Abstract

In ontology-mediated query answering, access to incomplete data sources is mediated by a conceptual layer constituted by an ontology. To correctly compute answers to queries, it is necessary to perform complex reasoning over the constraints expressed by the ontology. In the literature, there exists a multitude of techniques incorporating the ontological knowledge into queries. However, few of these approaches were designed for comprehensibility of the query answers. In this article, we try to bridge these two qualities by adapting a proof framework originally applied to axiom entailment for conjunctive query answering. We investigate the data and combined complexity of determining the existence of a proof below a given quality threshold, which can be measured in different ways. By distinguishing various parameters such as the shape of a query, we obtain an overview of the complexity of this problem for the lightweight ontology languages *DL-Lite*$_R$,and also have a brief look at temporal query answering.

1. Introduction

Explaining description logic (DL) reasoning has a long tradition, starting with the first works on *proofs* for standard DL entailments [1, 2]. A popular and very effective method is *justifications*, which simply point out the axioms from an ontology that are responsible for an entailment [3, 4, 5, 6]. More recently, work has resumed on techniques to find proofs for explaining more complex logical consequences [7, 8, 9, 10, 11]. On the other hand, if a desired entailment does not hold, one needs different explanation techniques such as abduction [12, 13, 14] or counterinterpretations [15]. Explaining answers to conjunctive queries (CQs) has also been investigated before, in the form of abduction for missing answers over *DL-Lite* ontologies [14], provenance for positive answers in *DL-Lite* and $\mathcal{EL}$ [16, 17], as well as proofs for *DL-Lite* query answering [18, 19, 20].

Here, we also investigate proofs for CQ answers, inspired by [18, 19, 20], but additionally consider the problem of generating *good* proofs according to some quality measures and provide a range of complexity results focussing on *DL-Lite*$_R$.In addition to classical OMQA, we also have a brief look at explaining inferences over temporal data using a query language incorporating metric temporal operators. Our results are based on a framework developed for proofs of standard DL reasoning [9]. There, proofs are formalized as directed, acyclic hypergraphs and

DL 2022: 35th International Workshop on Description Logics, August 7–10, 2022, Haifa, Israel

christian.alrabbaa@tu-dresden.de (C. Alrabbaa); stefan.borgwardt@tu-dresden.de (S. Borgwardt); patrick.koopmann@tu-dresden.de (P. Koopmann); alisa.kovtunova@tu-dresden.de (A. Kovtunova)
0000-0002-2925-1765 (C. Alrabbaa); 0000-0003-0924-8478 (S. Borgwardt); 0000-0001-5999-2583 (P. Koopmann); 0000-0001-9936-0943 (A. Kovtunova)

proof quality can be measured in different ways. We mainly consider the *size* (the number of formulas) of a proof as well as its *tree size*, which corresponds to the size when the proof is presented in a tree-shaped way (which may require repeating subproofs), as it is often done in practice [8, 21]. The quest for good proofs is formalized as a search problem in a so-called *derivation structures* produced by a *deriver*, which specifies the possible inferences.

In this paper, we consider two different kinds of derivers for generating proofs for CQ answers. These loosely correspond to the approaches in [18, 19, 20], but are generalized to apply to a larger class of DLs. Specifically, our structures rely on a translation of DLs to *existential rules* [22], and thus apply to any DL that can be expressed in this formalism. One deriver, denoted by $\mathfrak{D}_{\mathsf{cq}}$ and inspired by [19, 20], focuses on the derivation of CQs, which can be derived from other CQs and ontology axioms. Inferences in $\mathfrak{D}_{\mathsf{cq}}$ are logically sound, but can be harder to understand. The reason is the local scope of existential quantification in a CQ, which forces atoms connected by the same variables to be carried along inferences they are not relevant for. This problem is circumvented with the deriver $\mathfrak{D}_{\mathsf{sk}}$, which relies on a Skolemized version of the TBox. This allows one to focus on inferences of single atoms that are only later aggregated into the final CQ, leading to simpler sentences within the proof. Focusing on the particular cases of *DL-Lite*$_R$ and $\mathcal{EL}$, we consider the complexity of the decision problems of finding proofs of (tree) size below a given threshold n in these derivation structures. We find that for *DL-Lite*$_R$ and any DL in which CQ answering is UCQ-rewritable, all of these problems (regardless of derivation structure and quality measure) are in AC^0 in data complexity. In combined complexity, these problems are NP-complete in general, but polynomial when considering only acyclic queries and tree size. We also obtain similar results for the case of $\mathfrak{D}_{\mathsf{sk}}$ w.r.t. $\mathcal{EL}$ ontologies and tree size, but for size the situation is not clear yet because we suspect that for $\mathcal{EL}$ proofs may actually get exponentially large. To explain answers to *temporal* queries, we extend our derivers with new inference schemes to deal with metric temporal operators, allowing us to lift some of our results also to this setting. The full details can be found in a technical report [23], but we describe the main ideas here.

2. Preliminaries

Proofs In our setting, a *logic* $\mathcal{L} = (\mathcal{S}_\mathcal{L}, \models_\mathcal{L})$ consists of a set $\mathcal{S}_\mathcal{L}$ of *$\mathcal{L}$-sentences* and a *consequence relation* $\models_\mathcal{L} \subseteq P(\mathcal{S}_\mathcal{L}) \times \mathcal{S}_\mathcal{L}$ between *$\mathcal{L}$-theories* (subsets of $\mathcal{L}$-sentences) and single $\mathcal{L}$-sentences; we usually write only $\models$ instead of $\models_\mathcal{L}$. We assume that the *size* $|\eta|$ of an $\mathcal{L}$-sentence η is defined in some way, e.g. by the number of symbols in η. We require that $\mathcal{L}$ is *monotonic*, i.e. that $\mathcal{T} \models \eta$ implies $\mathcal{T}' \models \eta$ for all $\mathcal{T}' \supseteq \mathcal{T}$. For example, $\mathcal{L}$ could be *first-order logic* or some DL.

As in [9, 10, 11], we view proofs as directed hypergraphs (see the appendix for details).

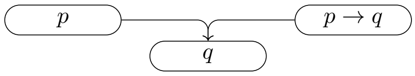
Definition 1 (Derivation Structure)**.** A *derivation structure* $\mathcal{D} = (V, E, \ell)$ over a theory $\mathcal{U}$ is a directed, labeled hypergraph that is

- *grounded*, i.e. every leaf v in $\mathcal{D}$ is labeled by $\ell(v) \in \mathcal{U}$; and
- *sound*, i.e. for every hyperedge $(S, d) \in E$, the entailment $\{\ell(s) \mid s \in S\} \models \ell(d)$ holds.

We call hyperedges $(S, d) \in E$ *inferences* or *inference steps*, with S being the *premises* and d the *conclusion*, and may write them like

$$\frac{p \qquad p \to q}{q}$$

or



Proofs are special derivation structures that derive a goal sentence.

Definition 2 (Proof). Given a sentence η and a theory $\mathcal{U}$, a *proof of* $\mathcal{U} \models \eta$ is a finite derivation structure $\mathcal{P} = (V, E, \ell)$ over $\mathcal{U}$ such that

- $\mathcal{P}$ contains exactly one sink $v_\eta \in V$, which is labeled by η,
- $\mathcal{P}$ is acyclic, and
- every vertex has at most one incoming hyperedge, i.e. there exist no two hyperedges $(S_1, v), (S_2, v) \in E$ with $S_1 \neq S_2$.

A *tree proof* is a proof that is a tree. A *subproof* S of a hypergraph H is a subgraph of H that is a proof with $\mathsf{leaf}(S) \subseteq \mathsf{leaf}(H)$.

To compute proofs, we assume that there is some reasoning system or calculus that defines a derivation structure for a given entailment η, and the structure may contain several proofs for that entailment. Formally, a *deriver* $\mathfrak{D}$ for a logic $\mathcal{L}$ takes as input an $\mathcal{L}$-theory $\mathcal{U}$ and an $\mathcal{L}$-sentence η, and returns a (possibly infinite) derivation structure $\mathfrak{D}(\mathcal{U}, \eta)$ over $\mathcal{U}$ that describes all inference steps that $\mathfrak{D}$ could perform in order to derive η from $\mathcal{U}$. This derivation structure is not necessarily computed explicitly, but can be accessed through an oracle (which checks, for example, whether an inference conforms to the underlying calculus). The task of finding a good proof then corresponds to finding a (finite) proof that can be homomorphically mapped into this derivation structure and which is minimal according to some measure of proof quality. We consider two such measures here: the *size* of a proof $\mathcal{P} = (V, E, \ell)$ is $\mathfrak{m}_\mathsf{s}(\mathcal{P}) := |V|$,[1] and the *tree size* $\mathfrak{m}_\mathsf{t}(\mathcal{P})$ is the size of a tree unraveling of $\mathcal{P}$ [11]. The *depth* of $\mathcal{P}$ is the length of the longest path from a leaf to the sink (see appendix).

DLs and Existential Rules We assume that the reader is familiar with DLs, in particular *DL-Lite*$_R$ [24] and $\mathcal{EL}$ [25], where theories $\mathcal{U} = \mathcal{T} \cup \mathcal{A}$ are called *ontologies* or *knowledge bases* and are composed of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. Many DL ontologies can be equivalently expressed using the formalism of existential rules [22]. Existential rules are first-order sentences of the form $\forall \vec{y}, \vec{z}.\, \psi(\vec{y}, \vec{z}) \to \exists \vec{u}.\, \chi(\vec{z}, \vec{u})$, with the *body* $\psi(\vec{y}, \vec{z})$ and the *head* $\chi(\vec{z}, \vec{u})$ being conjunctions of atoms of the form $A(x)$ or $P(x_1, x_2)$, for a concept name A, role name P and terms x, x_1 and x_2, which are individual names or variables from $\vec{z}$, $\vec{u}$ and $\vec{y}$. We usually omit the universal quantification. Notable DLs that can be equivalently expressed as sets of existential rules are $\mathcal{EL}$, *Horn-*$\mathcal{SRIQ}$ and *DL-Lite*$_R$.

[1] Since every vertex has at most one incoming hyperedge, the size of E is at most quadratic in $|V|$.

Conjunctive Queries In this paper, we want to construct proofs for ontology-mediated conjunctive query entailments. A *conjunctive query (CQ)* $\mathbf{q}(\vec{x})$ is an expression of the form $\exists\vec{y}.\,\phi(\vec{x},\vec{y})$, where $\phi(\vec{x},\vec{y})$ is a conjunction of atoms using *answer variables* $\vec{x}$ and *existentially quantified variables* $\vec{y}$. If $\vec{x}=()$, then $\mathbf{q}(\vec{x})$ is called *Boolean.* ABox assertions are a special case of Boolean CQs with only one atom and no variables. A tuple $\vec{a}$ of individual names from $\mathcal{A}$ is a *certain answer* to $\mathbf{q}(\vec{x})$ over $\mathcal{T}\cup\mathcal{A}$, in symbols $\mathcal{T}\cup\mathcal{A}\models\mathbf{q}(\vec{a})$, if, for any model of $\mathcal{T}\cup\mathcal{A}$, the sentence $\mathbf{q}(\vec{a})$ is true in this model. Any CQ $\mathbf{q}(\vec{x})=\exists\vec{y}.\,\phi(\vec{x},\vec{y})$ is associated with the set of atoms in ϕ, so we can write e.g. $A(z)\in\mathbf{q}(\vec{x})$.

Example 1. For the following *DL-Lite$_R$* ontology and query, we have $\mathcal{T}\cup\{B(\mathsf{b})\}\models\mathbf{q}(\mathsf{b})$.

$$\mathcal{T}=\{A\sqsubseteq\exists R,\qquad \exists R^-\sqsubseteq\exists T,\qquad B\sqsubseteq\exists P,\qquad \exists P^-\sqsubseteq\exists S,\qquad P\sqsubseteq R^-\}$$
$$\mathbf{q}(y'')=\exists x,x',x'',y,y',z,z'.\,R(x,y)\wedge T(y,z)\wedge T(y',z)\wedge R(x',y')\wedge S(x',z')\wedge S(x'',z')\wedge P(y'',x'').$$

In the next section, we explore different ways to explain this inference (see Figures 2 and 4).

3. Derivation Structures for Certain Answers

In the following, let $\mathcal{T}\cup\mathcal{A}$ be a knowledge base in some DL $\mathcal{L}$, $\mathbf{q}$ a conjunctive query, and $\vec{a}$ a certain answer, i.e. $\mathcal{T}\cup\mathcal{A}\models\mathbf{q}(\vec{a})$, which we want to explain. We can use derivation structures over $\mathcal{L}_{cq}$ (the extension of $\mathcal{L}$ with all Boolean CQs) to explain query answers. For example, the following derivation step involving the ontology from Example 1 is a sound inference:

$$\frac{B(\mathsf{b})\qquad\mathcal{T}}{\mathbf{q}(\mathsf{b})}$$

However, to define a derivation structure that yields proofs suitable for explanations to users, inferences that only make small deduction steps are more valuable. For this purpose, we define derivers that capture which inference steps are admitted. For TBox entailment, in [9, 10, 11], we considered derivers based on the inference schemas used by a consequence-based reasoner. To obtain proofs for CQ entailment, we follow the ideas of *chase* procedures that replace atoms in CQs by other atoms by “applying” rules to them [26, 22, 24, 18]. We will introduce two derivers that represent different paradigms of what constitutes a proof.

3.1. The CQ Deriver

Similarly to the approach used in [19, 20], inferences in our first deriver, $\mathfrak{D}_{\mathsf{cq}}$, always produce Boolean CQs. This deriver is defined for DLs that can be expressed using existential rules. An inference step is obtained by matching the left-hand side of a rule to part of a CQ and then replacing it by the right-hand side. For example, starting from $\exists z.\,P(\mathsf{b},z)$ and $P(x,y)\to R(y,x)$, we can apply the substitution $\{x\mapsto\mathsf{b},y\mapsto z\}$ to obtain $\exists z.\,R(z,\mathsf{b})$. Additionally, we allow to keep any of the replaced atoms from the original CQ, e.g. to produce the conclusion $\exists z.\,P(\mathsf{b},z)\wedge R(z,\mathsf{b})$. A second type of inference allows one to combine two Boolean CQs using conjunction. To duplicate variables, we additionally introduce tautological rules such as $P(x,z)\to\exists z'.\,P(x,z')$,

$$\frac{\exists \vec{x}.\, \phi(\vec{x}) \qquad \psi(\vec{y},\vec{z}) \rightarrow \exists \vec{u}.\, \chi(\vec{z},\vec{u})}{\exists \vec{w}.\rho(\vec{w})}\ \textbf{(MP)} \qquad \frac{\exists \vec{x}.\, \phi(\vec{x}) \qquad \exists \vec{y}.\, \psi(\vec{y})}{\exists \vec{x},\vec{y}'.\phi(\vec{x}) \wedge \psi(\vec{y}')}\ \textbf{(C)}$$

$$\frac{}{\phi(\vec{x},\vec{y}) \rightarrow \exists \vec{x}.\, \phi(\vec{x},\vec{y})}\ \textbf{(T)} \qquad \frac{\exists \vec{x}.\, \phi(\vec{x},\vec{a})}{\exists \vec{x},\vec{y}.\, \phi(\vec{x},\vec{y})}\ \textbf{(E)}$$

Figure 1: Inference schemas for $\mathfrak{D}_{\mathsf{cq}}$. **(MP)** and **(T)** refer to *modus ponens* and *tautology*.

which yields $\exists z, z'.\, P(\mathrm{b}, z) \wedge P(\mathrm{b}, z')$ when combined with $\exists z.\, P(\mathrm{b}, z)$. Finally, we use an inference schema that allows us to replace constants by variables, e.g. to capture that $\exists z.\, P(\mathrm{b}, z)$ implies $\exists x, z.\, P(x, z)$.

The detailed inference schemas can be found in Figure 1. **(MP)** is admissible only if there exists a substitution π such that $\pi(\psi(\vec{y},\vec{z})) \subseteq \phi(\vec{x})$, and then $\rho(\vec{w})$ is the result of replacing *any subset of* $\pi(\psi(\vec{y},\vec{z}))$ in $\phi(\vec{x})$ by *any subset of* $\pi(\chi(\vec{z},\vec{u}'))$, where the variables $\vec{u}$ are renamed into new existentially quantified variables $\vec{u}'$ to ensure that they are disjoint with $\vec{x}$. In **(C)**, we again rename the variables $\vec{y}$ to $\vec{y}'$ to avoid overlap with $\vec{x}$. Since every ABox assertion corresponds to a ground CQ, this inference also allows one to collect ABox assertions into a single CQ. **(T)** introduces an existential rule that allows us, together with **(MP)**, to create copies of variables in CQs (see Fig. 2). Finally, **(E)** transforms individual names in some positions into existentially quantified variables.

Definition 3 (CQ Deriver)**.** $\mathfrak{D}_{\mathsf{cq}}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ is a derivation structure over $\mathcal{T} \cup \mathcal{A}$ with vertices labeled by the axioms in $\mathcal{T} \cup \mathcal{A}$ and all Boolean CQs over the signature of $\mathcal{T} \cup \mathcal{A}$, and its hyperedges represent all possible instances of **(MP)**, **(C)**, **(T)**, and **(E)** over these vertices. An *(admissible) proof in* $\mathfrak{D}_{\mathsf{cq}}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ is a proof of $\mathcal{T} \cup \mathcal{A} \models \mathbf{q}(\vec{a})$ that has a homomorphism into this derivation structure.

It is easy to check that the inferences used by $\mathfrak{D}_{\mathsf{cq}}$ are sound. Moreover, we can show that they are complete, i.e. that any CQ entailed by $\mathcal{A} \cup \mathcal{T}$ has a proof in $\mathfrak{D}_{\mathsf{cq}}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ (see Lemma 5). A proof for Example 1 w.r.t. $\mathfrak{D}_{\mathsf{cq}}$ is depicted in Figure 2.

3.2. Skolemized Derivation Structure

To explain a Boolean CQ, using a derivation structure that works on CQs seems natural. However, a downside is that we have to "collect" quantified variables along the proof and label vertices with complex expressions. Since the inference rules apply on sub-expressions, it may be challenging to understand on which part of the CQ an inference is performed—indeed, finding a match for the body o a rule in a CQ is NP-hard. The problem is that we cannot separate inference steps on the same variable without affecting soundness, as the existential quantification only applies locally in the current CQ. To follow our example: x'' and z' in Figure 2 are connected to each other and to the constant b, and thus have to be kept together: although $\exists x'', z'.P(\mathrm{b}, x'') \wedge S(x'', z')$ implies $\exists x''.P(\mathrm{b}, x'')$ and $\exists x'', z'.S(x'', z')$, those two CQs do not imply the original CQ anymore. To overcome these issues, we consider a second type of deriver that relies on Skolemization, and is inspired by the approach from [18].

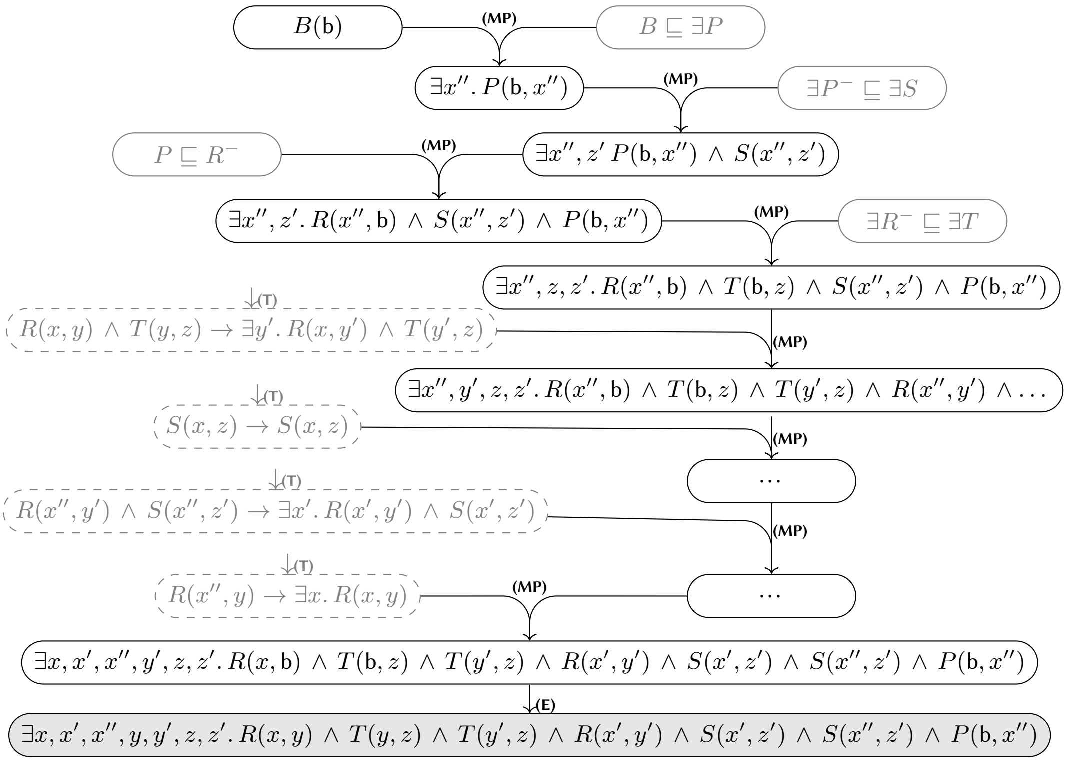


Figure 2: A CQ proof for Example 1 (inferences **(E)** and **(T)** are delayed to the last steps)

This deriver, $\mathfrak{D}_{\mathsf{sk}}$, mainly operates on ground CQs, and requires the theory to be *Skolemized*. This means that it cannot contain existential quantification, it may however contain function symbols. To Skolemize existential rules, for each existentially quantified variable a fresh function symbol is introduced; for the CI $\exists P^- \sqsubseteq \exists S$ this results in $P(x, y) \to S(y, g(y))$, where g is a unary function symbol whose argument denotes the dependency on the variable y shared between the body and head of the rule. Let $\mathcal{T}^s$ be the set of Skolemized rules resulting from this transformation and note that the entailments $\mathcal{T} \cup \mathcal{A} \models \mathbf{q}(\vec{a})$ and $\mathcal{T}^s \cup \mathcal{A} \models \mathbf{q}(\vec{a})$ are equivalent for CQs $\mathbf{q}(\vec{x})$ that do not use function symbols. Our deriver internally considers two kinds of formulas: 1) CQs that may use function symbols and 2) rules of the form $\forall \vec{x}.\phi(\vec{x}) \to \psi(\vec{x})$, where $\psi(\vec{x})$ may now contain function terms, but no further quantified variables. Since we are only interested in CQs that are entailed by $\mathcal{T}^s \cup \mathcal{A}$, we can assume w.l.o.g. that this entailment can be shown solely using domain elements denoted by ground terms, e.g. $f(f(\mathsf{a}))$, which allows us to eliminate variables from most of the inferences. For example, instead of $\exists x'', z'. P(\mathsf{b}, x'') \wedge S(x'', z')$ in Figure 2 we now use $P(\mathsf{b}, f(\mathsf{b})) \wedge S(f(\mathsf{b}), g(f(\mathsf{b})))$. Since these atoms do not share variables, in our derivation structure we mainly need to consider inferences on single atoms, which allows for more fine-grained proofs (see Figure 4). Only at the end we need to compose atoms to obtain a CQ.

The simplified inference schemas are shown in Figure 3. In $(\mathbf{MP_s})$, $\alpha_i(\vec{t}_i)$ and $\beta(\vec{s})$ are ground atoms with terms composed from individual names and Skolem functions, and likewise $\chi(\vec{z})$

$$\frac{\alpha_1(\vec{t}_1) \quad \dots \quad \alpha_n(\vec{t}_n) \quad \psi(\vec{y},\vec{z}) \to \chi(\vec{z})}{\beta(\vec{s})}\ \textbf{(MP}_\textbf{s}\textbf{)}$$

$$\frac{\alpha_1(\vec{t}_1) \quad \dots \quad \alpha_n(\vec{t}_n)}{\alpha_1(\vec{t}_1) \wedge \dots \wedge \alpha_n(\vec{t}_n)}\ \textbf{(C}_\textbf{s}\textbf{)} \qquad \frac{\phi(\vec{t})}{\exists \vec{x}.\phi(\vec{x})}\ \textbf{(E}_\textbf{s}\textbf{)}$$

Figure 3: Inference schemas for $\mathfrak{D}_{\mathsf{sk}}$.

may contain Skolem functions; similar to **(MP)**, we require that there is a substitution π such that $\pi(\psi(\vec{y},\vec{z})) = \{\alpha_1(\vec{t}_1),\dots,\alpha_n(\vec{t}_n)\}$ and $\beta(\vec{s}) \in \pi(\chi(\vec{z}))$. In $\textbf{(E}_\textbf{s}\textbf{)}$, $\vec{t}$ is now a vector of ground terms which may contain function symbols. Since $\textbf{(MP}_\textbf{s}\textbf{)}$ works only with ground atoms, $\textbf{(C}_\textbf{s}\textbf{)}$ and $\textbf{(E}_\textbf{s}\textbf{)}$ can now only be used at the end of a proof to obtain the desired CQ (see Figure 4). Moreover, we do not need a version of **(T)** here since it would be trivial for ground atoms. Its effects in $\mathfrak{D}_{\mathsf{cq}}$ can be simulated here due to the fact that the same atom can be used several times as a premise for $\textbf{(MP}_\textbf{s}\textbf{)}$ or $\textbf{(C}_\textbf{s}\textbf{)}$.

Definition 4 (Skolemized Deriver)**.** The derivation structure $\mathfrak{D}_{\mathsf{sk}}(\mathcal{T}^s \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ is defined similarly to Definition 3, but using $\mathcal{T}^s$ and the inference schemas $\textbf{(MP}_\textbf{s}\textbf{)}$, $\textbf{(C}_\textbf{s}\textbf{)}$ and $\textbf{(E}_\textbf{s}\textbf{)}$.

Though different presentations with different advantages and disadvantages, it is not hard to translate proofs based on $\mathfrak{D}_{\mathsf{sk}}$ into proofs in $\mathfrak{D}_{\mathsf{cq}}$ and vice versa.

Lemma 5. *Any proof $\mathcal{P}$ in $\mathfrak{D}_{\mathsf{cq}}(\mathcal{T}\cup\mathcal{A}, \mathbf{q}(\vec{a}))$ can be transformed into a proof in $\mathfrak{D}_{\mathsf{sk}}(\mathcal{T}^s\cup\mathcal{A}, \mathbf{q}(\vec{a}))$ in time polynomial in the sizes of $\mathcal{P}$ and $\mathcal{T}$, and conversely any proof $\mathcal{P}$ in $\mathfrak{D}_{\mathsf{sk}}(\mathcal{T}^s \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ can be transformed into a proof in $\mathfrak{D}_{\mathsf{cq}}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ in time polynomial in the sizes of $\mathcal{P}$ and $\mathcal{T}$. The latter also holds for tree proofs.*

However, it is not the case that *minimal* proofs are equivalent for these two derivers, i.e. a minimal proof may become non-minimal after the transformation.

This lemma also shows that our derivation structures are complete, i.e. if $\mathcal{T}\cup\mathcal{A} \models \mathbf{q}(\vec{a})$ holds, then we can provide a proof for it. To see this, consider the minimal Herbrand model H of $\mathcal{T}^s\cup\mathcal{A}$, which can be computed using the *(Skolem) chase* procedure for existential rules—essentially, applying the rules step-by-step to obtain new ground atoms, in a way very similar to $\textbf{(MP}_\textbf{s}\textbf{)}$. This model is a universal model for CQ answering over $\mathcal{T}\cup\mathcal{A}$, which means that $\mathcal{T}\cup\mathcal{A} \models \mathbf{q}(\vec{a})$ implies $H \models \mathbf{q}(\vec{a})$, which, in turn, means that there must be a proof in $\mathfrak{D}_{\mathsf{sk}}(\mathcal{T}^s \cup \mathcal{A}, \mathbf{q}(\vec{a}))$, and hence by Lemma 5 also one in $\mathfrak{D}_{\mathsf{cq}}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$. For convenience, we assume in the following that TBoxes are silently Skolemized when constructing derivation structures using $\mathfrak{D}_{\mathsf{sk}}$, that is, we identify $\mathfrak{D}_{\mathsf{sk}}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ with $\mathfrak{D}_{\mathsf{sk}}(\mathcal{T}^s \cup \mathcal{A}, \mathbf{q}(\vec{a}))$.

4. The Complexity of Finding Good Proofs

It is our intution that proofs in $\mathfrak{D}_{\mathsf{sk}}$ are more comprehensible than in $\mathfrak{D}_{\mathsf{cq}}$ because of its simpler labels. Moreover, we assume *small* proofs (w.r.t. size $\mathfrak{m}_{\mathsf{s}}$ or tree size $\mathfrak{m}_{\mathsf{t}}$) to be more comprehensible than large ones (but one can certainly also consider other measures [10, 11]). Therefore,

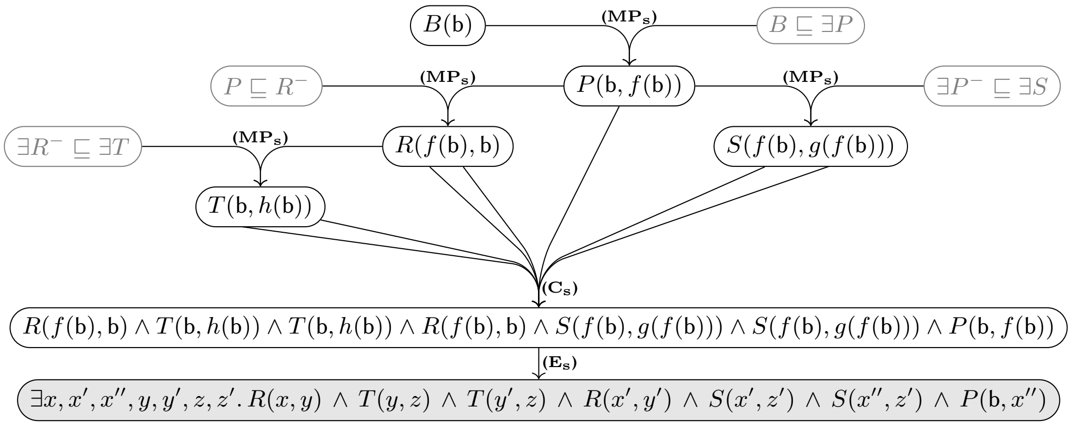


Figure 4: A Skolemized proof for Example 1

we now study the *complexity* of finding small proofs automatically (which is independent of the comprehensibility of the resulting proofs). More precisely, we are interested in the following decision problem $\mathsf{OP}_\mathsf{x}(\mathcal{L}, \mathfrak{m})$ for a deriver $\mathfrak{D}_\mathsf{x} \in \{\mathfrak{D}_\mathsf{cq}, \mathfrak{D}_\mathsf{sk}\}$, a DL $\mathcal{L} \in \{\mathcal{EL}, \textit{DL-Lite}_R\}$, and a measure $\mathfrak{m} \in \{\mathfrak{m}_\mathsf{s}, \mathfrak{m}_\mathsf{t}\}$: given an $\mathcal{L}$-KB $\mathcal{T} \cup \mathcal{A}$, a query $\mathbf{q}(\vec{x})$ with certain answer $\vec{a}$, and a natural number n (in binary encoding), is there a proof $\mathcal{P}$ for $\mathbf{q}(\vec{a})$ in $\mathfrak{D}_\mathsf{x}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ with $\mathfrak{m}(\mathcal{P}) \leq n$? To better distinguish the complexity of finding small proofs from that of query answering, we assume $\mathcal{T} \cup \mathcal{A} \models \mathbf{q}(\vec{a})$ as prerequisite, which fits the intuition that users request an explanation only after they know that $\vec{a}$ is a certain answer. Lemma 7 in [11] shows that, instead of looking for arbitrary proofs and homomorphisms into the derivation structure, one can restrict the search to subproofs of $\mathfrak{D}_\mathsf{x}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$, which we will often do implicitly.

It is common in the context of OMQA to distinguish between *data complexity*, where only the data varies, and *combined complexity*, where also the influence of the other inputs is taken into account. This raises the question whether the bound n is seen as part of the input or not. It turns out that fixing n trivializes the data complexity, because then n also fixes the set of relevant ABoxes modulo isomorphism.

Theorem 6. *For a constant bound n, $\mathsf{OP}_\mathsf{x}(\mathcal{L}, \mathfrak{m})$ is in AC^0 in data complexity.*

One may argue that, since the size of the proof depends on $\mathcal{A}$, the bound n on the proof size should be considered part of the data as well. Under this assumption, our decision problem is not necessarily in AC^0 anymore. For example, consider the $\mathcal{EL}$ TBox $\{\exists r.A \sqsubseteq A\}$ and $q(x) \leftarrow A(x)$. For every n, there is an ABox $\mathcal{A}$ such that $A(a)$ is entailed by a seqeuence of n role assertions, and thus needs a proof of size at least n. Deciding whether this query admits a bounded proof is thus as hard as deciding whether it admits an answer at all in $\mathcal{A}$, i.e. P-hard [27]. However, we at least stay in AC^0 for DLs over which CQs are rewritable, e.g. *DL-Lite*$_R$ [24], because the number of (non-isomorphic) proofs that we need to consider is bounded by the size of the rewriting, which is constant in data complexity.

Theorem 7. *If all CQs are UCQ-rewritable over $\mathcal{L}$-TBoxes, then $\mathsf{OP}_\mathsf{x}(\mathcal{L}, \mathfrak{m})$ is in AC^0 in data complexity.*

We now consider the combined complexity. In [9, 11], we established general upper bounds for finding proofs of bounded size. These results depend only on the size of the derivation structure obtained for the given input. Both $\mathfrak{D}_{\mathsf{cq}}$ and $\mathfrak{D}_{\mathsf{sk}}$ may produce derivation structures of infinite size, as $\mathfrak{D}_{\mathsf{cq}}$ contains CQs of arbitrary size, and $\mathfrak{D}_{\mathsf{sk}}$ also has Skolem terms of arbitrary nesting depth. However, we can sometimes bound the number of relevant Skolem terms in $\mathfrak{D}_{\mathsf{sk}}$ by considering only the part of the minimal Herbrand model H that is necessary to satisfy the query $\mathbf{q}(\vec{a})$. For example, in logics with the *polynomial witness property* [28], including *DL-Lite*$_R$, we know that any query that is entailed is already satisfied after polynomially many chase steps used to construct H. In particular, this means that the nesting depth of Skolem terms in a proof is bounded polynomially (in the size of the TBox and the query), and hence the part of $\mathfrak{D}_{\mathsf{sk}}(\mathcal{T}^s \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ that we need to search for a (small) proof is bounded exponentially. For such structures, our results from [9, 11] give us a NExpTime-upper bound for size, and a PSpace-upper bound for tree size, upon which we can improve with the following lemma.

Lemma 8. *There is a polynomial p such that for any DL-Lite$_R$ KB $\mathcal{T} \cup \mathcal{A}$, CQ $\mathbf{q}(\vec{x})$, and certain answer $\vec{a}$, there is a proof in $\mathfrak{D}_{\mathsf{sk}}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$ of tree size at most $p(|\mathcal{T}|, |\mathbf{q}(\vec{x})|)$.*

A direct consequence of Lemmas 5 and 8 is the upper bound in the following theorem. The lower bound can be shown by a reduction from Boolean query entailment over *DL-Lite*$_R$ ontologies: for this, we extend the KB in a given query answering problem by axioms that trivially entail the query, but only yield proofs larger than n.

Theorem 9. $\mathsf{OP}_{\mathsf{x}}(\textit{DL-Lite}_R, \mathfrak{m})$ *is NP-complete.*

To obtain tractability, we can restrict the shape of the query. Recall that the *Gaifman graph* of a query $\mathbf{q}$ is the undirected graph using the terms of $\mathbf{q}$ as nodes and has an edge between terms occurring together in an atom. A query is *tree-shaped* if its Gaifman graph is a tree.

Theorem 10. *Given a DL-Lite$_R$ KB $\mathcal{T} \cup \mathcal{A}$ and a tree-shaped CQ $\mathbf{q}(\vec{x})$ with certain answer $\vec{a}$, one can compute in polynomial time a proof of minimal tree size in $\mathfrak{D}_{\mathsf{sk}}(\mathcal{T} \cup \mathcal{A}, \mathbf{q}(\vec{a}))$.*

The central property used in the proof of Theorem 10 is that for tree size every atom in $\mathbf{q}(\vec{a})$ has a separate proof, even if two atoms are proven in the same way. To avoid this redundancy, one could think about modifying $(\mathbf{E_s})$ slightly:

$$\frac{\phi(\vec{t})}{\exists \vec{x}.\phi'(\vec{x})}\ (\mathbf{E'_s})\text{ , provided there exists } \sigma\colon \vec{x} \to \vec{t} \text{ s.t. } \phi'(\vec{x})\sigma = \phi(\vec{t})$$

Denote the resulting deriver by $\mathfrak{D}'_{\mathsf{sk}}$. Using $(\mathbf{E'_s})$, we can derive $\exists x, y.\ A(x) \wedge A(y)$ from $A(a)$; with $(\mathbf{E_s})$, the premise would need to be $A(a) \wedge A(a)$. However, this modification is already sufficient to make our problem NP-hard for tree-shaped queries, even *without a TBox.* The same problem arises in $\mathfrak{D}_{\mathsf{cq}}$ (where atoms can be duplicated using **(T)**), and if we consider $\mathfrak{m}_{\mathsf{s}}$.

Theorem 11. *For tree-shaped CQs,* $\mathsf{OP}'_{\mathsf{x}}(\mathcal{L}, \mathfrak{m}_{\mathsf{t}})$ *is NP-hard. The same holds for* $\mathsf{OP}_{\mathsf{sk}}(\mathcal{L}, \mathfrak{m}_{\mathsf{s}})$ *and* $\mathsf{OP}_{\mathsf{cq}}(\mathcal{L}, \mathfrak{m}_{\mathsf{t}})$.

Table 1
Semantics of (Boolean) MTCQs for $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_i)_{i\in\mathbb{Z}})$ and $i \in \mathbb{Z}$.

ϕ	$\mathfrak{I}, i \models \phi$ iff
CQ ψ	$\mathcal{I}_i \models \psi$
$\top$	true
$\phi \wedge \psi$	$\mathfrak{I}, i \models \phi$ and $\mathfrak{I}, i \models \psi$
$\phi \vee \psi$	$\mathfrak{I}, i \models \phi$ or $\mathfrak{I}, i \models \psi$
$\boxplus_I \phi$	$\forall k \in I$ such that $\mathfrak{I}, i + k \models \phi$
$\boxminus_I \phi$	$\forall k \in I$ such that $\mathfrak{I}, i - k \models \phi$
$\phi \mathcal{U}_I \psi$	$\exists k \in I$ such that $\mathfrak{I}, i+k \models \psi$ and $\forall j : 0 \leq j < k : \mathfrak{I}, i+j \models \phi$
$\phi \mathcal{S}_I \psi$	$\exists k \in I$ such that $\mathfrak{I}, i-k \models \psi$ and $\forall j : 0 \leq j < k : \mathfrak{I}, i-j \models \phi$

5. Metric Temporal CQs

We now consider proofs for *temporal* query answering. In this setting, TBox axioms hold globally, i.e. at all time points, the ABox contains information about the state of the world in different time intervals, and the query contains (metric) temporal operators.

An *interval* ι is a *nonempty* subset of $\mathbb{Z}$ of the form $[t_1, t_2]$, where $t_1, t_2 \in \mathbb{Z} \cup \{\infty\}$ and $t_1 \leq t_2$ (for simplicity, we write $[\infty, t_2]$ for $(-\infty, t_2]$ and $[t_1, \infty]$ instead of $[t_1, \infty)$);[2] t_1 and t_2 are encoded in binary. A temporal ABox $\mathcal{A}$ is a finite set of facts of the form $A(a)@\iota$ or $P(a, b)@\iota$, where $A(a)$ and $P(a, b)$ are assertions and ι is an interval. The fact $A(a)@\iota$ states that $A(a)$ holds throughout the interval ι. We denote by $\mathsf{tem}(\mathcal{A})$ the multiset of intervals that occur in $\mathcal{A}$ and $|\mathsf{tem}(\mathcal{A})|$ is the sum of their lengths. A *temporal interpretation* $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_i)_{i\in\mathbb{Z}})$, is a collection of DL interpretations $\mathcal{I}_i = (\Delta^{\mathfrak{I}}, \cdot^{\mathcal{I}_i})$, $i \in \mathbb{Z}$, over $\Delta^{\mathfrak{I}}$. $\mathfrak{I}$ *satisfies* a TBox axiom α if each $\mathcal{I}_i$, $i \in \mathbb{Z}$, satisfies α, and it satisfies a temporal assertion $\alpha@\iota$ if each $\mathcal{I}_i$, $i \in \iota$, satisfies α.

We use the finite-range positive version of metric temporal conjunctive queries (MTCQs) introduced in [29, 30], combining CQs with MTL operators [31, 32, 33].

Definition 12. An MTCQ is of the form $\mathbf{q}(\vec{x}, w) = \phi(\vec{x})@w$, where ϕ is built according to

$$\phi ::= \psi \mid \top \mid \phi \wedge \phi \mid \phi \vee \phi \mid \boxminus_I \phi \mid \boxplus_I \phi \mid \phi \mathcal{U}_I \phi \mid \phi \mathcal{S}_I \phi,$$

with w an interval variable, ψ a CQ, I a finite interval with non-negative endpoints, and $\vec{x}$ the free variables of all CQs in ϕ. A certain answer to $\mathbf{q}(\vec{x}, w)$ over $\mathcal{T} \cup \mathcal{A}$ is a pair $(\vec{a}, \iota)$ such that $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$, ι is an interval and, for any $t \in \iota$ and any model $\mathfrak{I}$ of $\mathcal{T} \cup \mathcal{A}$, we have $\mathfrak{I}, t \models \phi(\vec{a})$ according to Table 1. We denote this as $\mathcal{T} \cup \mathcal{A} \models \mathbf{q}(\vec{a}, \iota)$.

For temporal extensions of Definitions 3 and 4, we will interpret $A \sqsubseteq A'$ now as the global temporal rule $A(x) \to A'(x)$ holding in any possible interval.

$$\frac{(\exists \vec{x}.\, \phi(\vec{x}))@\iota \qquad \psi(\vec{y}, \vec{z}) \to \exists \vec{u}.\, \chi(\vec{z}, \vec{u})}{(\exists \vec{w}.\rho(\vec{w}))@\iota} \ \textbf{(TMP)}$$

Similarly, we need temporal versions of **(C)** and **(E)**, where all CQs are annotated with the same interval variable. In addition, we need an inference for disjunctive MTCQS:

[2] This allows us to avoid considering special cases in the interval arithmetic below.

$$\frac{\phi(\vec{x})@\iota}{(\phi(\vec{x}) \vee \psi(\vec{y}))@\iota} \textbf{ (DISJ)}$$

To provide a proof for a *temporal* query, we need to be able to coalesce, i.e. merge intervals:

$$\frac{\exists\vec{x}_1.\,\phi(\vec{x}_1)@\iota_1 \qquad \dots \qquad \exists\vec{x}_n.\,\phi(\vec{x}_n)@\iota_n}{(\exists\vec{x}.\,\phi(\vec{x}))@\bigcup_{i=1}^{n}\iota_i} \textbf{ (COAL)}$$

where $\bigcup_{i=1}^{s}\iota_i$ is a single interval and $\phi(\vec{x}_1),\dots,\phi(\vec{x}_n)$ are identical up to variable renaming. On the other hand, we also need an inverse operation to shrink intervals:

$$\frac{\exists\vec{x}.\,\phi(\vec{x})@\iota}{\exists\vec{x}.\,\phi(\vec{x})@\iota'} \textbf{ (SEP)}$$

where $\iota' \subseteq \iota$. Both inferences are needed to infer all intervals ι with $\mathcal{T} \cup \mathcal{A} \models \exists\vec{x}.\,\phi(\vec{x})@\iota$.

Finally, we need inferences for the temporal operators, where for $\mathcal{U}_{[r_1,r_2]}$ we only consider the case where $r_1 > 0$ since $\phi\,\mathcal{U}_{[0,r_2]}\psi$ is equivalent to $\psi \vee (\phi\,\mathcal{U}_{[1,r_2]}\psi)$:

$$\frac{\phi(\vec{x})@[t_1,t_2]}{\boxplus_{[r_1,r_2]}\phi(\vec{x})@[t_1-r_1,t_2-r_2]} \textbf{ (}\boxplus\textbf{)} \qquad \frac{\phi(\vec{x})@\iota \qquad \psi(\vec{y})@\iota'}{\phi(\vec{x})\,\mathcal{U}_{[r_1,r_2]}\psi(\vec{y})@(\nu-[r_1,r_2])\cap\iota} \textbf{ (}\mathcal{U}\textbf{)}$$

where $\nu := (\iota+1)\cap\iota'$ (all time points where ψ-s are immediately preceded by ϕ-s) and $[w_1,w_2]-[r_1,r_2] := [w_1-r_2, w_2-r_1]$, and none of the involved intervals should be empty. Inferences for $\boxminus$ and $\mathcal{S}$ are similar. We denote the resulting deriver by $\mathfrak{D}_{\mathsf{tcq}}$. A Skolemized variant $\mathfrak{D}_{\mathsf{tsk}}$ can be defined similarly with temporalized versions of **(MP$_\mathbf{s}$)**, **(C$_\mathbf{s}$)**, and **(E$_\mathbf{s}$)**. We can now lift Theorems 7 and 9 to this setting.

Theorem 13. *If CQ answering in $\mathcal{L}$ is UCQ-rewritable, then MTCQ answering is also UCQ-rewritable and* $\mathsf{OP}_{\mathsf{tx}}(\mathcal{L},\mathfrak{m})$ *is in* AC^0 *in data complexity. Moreover,* $\mathsf{OP}_{\mathsf{tx}}(\textit{DL-Lite}_R,\mathfrak{m})$ *is NP-complete. Let* $\mathfrak{D} \in \{\mathfrak{D}_{\mathsf{tcq}},\mathfrak{D}_{\mathsf{tsk}}\}$*. Then, it is NP-complete to decide whether, given a DL-Lite$_R$ TBox $\mathcal{T}$, a temporal ABox $\mathcal{A}$, $\mathbf{q}(\vec{a},\iota)$ s.t. $\mathcal{T}\cup\mathcal{A} \models \mathbf{q}(\vec{a},\iota)$, and n in unary or binary encoding, there exists a proof in $\mathfrak{D}(\mathcal{T}\cup\mathcal{A},\mathbf{q}(\vec{a},\iota))$ of (tree) size at most n.*

6. Conclusion

We started to explore a framework for proofs of answers to conjunctive queries. In the future, we want to extend our complexity results to other DLs, and our framework to DLs that cannot be translated to existential rules. Other interesting research questions include derivers that combine TBox and query entailment rules, e.g. $\mathfrak{D}_{\mathsf{cq}}$ plus the rules of the ELK reasoner [34]. Instead of proofs, one could also try to show a canonical model to a user in order to explain query answers. For explaining missing answers, we also want to continue investigating how to find (optimal) counter-interpretations or abduction results [12].

Acknowledgments

This work was supported by DFG in grant 389792660, TRR 248 (https://perspicuous-computing.science), and QuantLA, GRK 1763 (https://lat.inf.tu-dresden.de/quantla).

References

[1] D. L. McGuinness, Explaining Reasoning in Description Logics, Ph.D. thesis, Rutgers University, NJ, USA, 1996. doi:10.7282/t3-q0c6-5305.

[2] A. Borgida, E. Franconi, I. Horrocks, Explaining $\mathcal{ALC}$ subsumption, in: ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, 2000, pp. 209–213. URL: http://www.frontiersinai.com/ecai/ecai2000/pdf/p0209.pdf.

[3] S. Schlobach, R. Cornet, Non-standard reasoning services for the debugging of description logic terminologies., in: G. Gottlob, T. Walsh (Eds.), Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003), Morgan Kaufmann, 2003, pp. 355–362. URL: http://ijcai.org/Proceedings/03/Papers/053.pdf.

[4] F. Baader, R. Peñaloza, B. Suntisrivaraporn, Pinpointing in the description logic $\mathcal{EL}^{+}$, in: KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI, KI 2007, 2007, pp. 52–67. doi:10.1007/978-3-540-74565-5_7.

[5] R. Peñaloza, Axiom-Pinpointing in Description Logics and Beyond, Ph.D. thesis, Technische Universität Dresden, Germany, 2009. URL: https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa-24743.

[6] M. Horridge, Justification Based Explanation in Ontologies, Ph.D. thesis, University of Manchester, UK, 2011. URL: https://www.research.manchester.ac.uk/portal/files/54511395/FULL_TEXT.PDF.

[7] M. Horridge, B. Parsia, U. Sattler, Justification oriented proofs in OWL, in: The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Part I, 2010, pp. 354–369. doi:10.1007/978-3-642-17746-0_23.

[8] Y. Kazakov, P. Klinov, A. Stupnikov, Towards reusable explanation services in protege, in: A. Artale, B. Glimm, R. Kontchakov (Eds.), Proc. of the 30th Int. Workshop on Description Logics (DL'17), volume 1879 of *CEUR Workshop Proceedings*, 2017. URL: http://www.ceur-ws.org/Vol-1879/paper31.pdf.

[9] C. Alrabbaa, F. Baader, S. Borgwardt, P. Koopmann, A. Kovtunova, Finding small proofs for description logic entailments: Theory and practice, in: E. Albert, L. Kovacs (Eds.), LPAR-23: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, volume 73 of *EPiC Series in Computing*, EasyChair, 2020, pp. 32–67. doi:10.29007/nhpp.

[10] C. Alrabbaa, F. Baader, S. Borgwardt, P. Koopmann, A. Kovtunova, On the complexity of finding good proofs for description logic entailments, in: S. Borgwardt, T. Meyer (Eds.), Proceedings of the 33rd International Workshop on Description Logics (DL'20), volume 2663 of *CEUR Workshop Proceedings*, 2020. URL: http://ceur-ws.org/Vol-2663/paper-1.pdf.

[11] C. Alrabbaa, F. Baader, S. Borgwardt, P. Koopmann, A. Kovtunova, Finding good proofs for description logic entailments using recursive quality measures, in: A. Platzer, G. Sutcliffe (Eds.), Proceedings of the 28th International Conference on Automated Deduction (CADE'21), volume 12699 of *LNCS*, 2021, pp. 291–308. doi:10.1007/978-3-030-79876-5_17.

[12] P. Koopmann, Signature-based abduction with fresh individuals and complex concepts for description logics, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, ijcai.org, 2021, pp. 1929–1935. doi:10.

24963/ijcai.2021/266.
[13] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicenavicius, Explanations for negative query answers under existential rules, in: D. Calvanese, E. Erdem, M. Thielscher (Eds.), Proceedings of KR 2020, AAAI Press, 2020, pp. 223–232. URL: https://doi.org/10.24963/kr.2020/23. doi:10.24963/kr.2020/23.
[14] D. Calvanese, M. Ortiz, M. Simkus, G. Stefanoni, The complexity of explaining negative query answers in dl-lite, in: G. Brewka, T. Eiter, S. A. McIlraith (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, AAAI Press, 2012. URL: http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4537.
[15] C. Alrabbaa, W. Hieke, A. Turhan, Counter model transformation for explaining non-subsumption in EL, in: C. Beierle, M. Ragni, F. Stolzenburg, M. Thimm (Eds.), Proceedings of the 7th Workshop on Formal and Cognitive Reasoning, volume 2961 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 9–22. URL: http://ceur-ws.org/Vol-2961/paper_2.pdf.
[16] D. Calvanese, D. Lanti, A. Ozaki, R. Peñaloza, G. Xiao, Enriching ontology-based data access with provenance, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, ijcai.org, 2019, pp. 1616–1623. doi:10.24963/ijcai.2019/224.
[17] C. Bourgaux, A. Ozaki, R. Peñaloza, L. Predoiu, Provenance for the description logic elhr, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, ijcai.org, 2020, pp. 1862–1869. doi:10.24963/ijcai.2020/258.
[18] A. Borgida, D. Calvanese, M. Rodriguez-Muro, Explanation in the *DL-Lite* family of description logics, in: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems: OTM 2008, volume 5332 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 1440–1457. doi:10.1007/978-3-540-88873-4_35.
[19] G. Stefanoni, Explaining query answers in lightweight ontologies, Diploma thesis, Technische Universität Wien, 2011. URL: http://www.cs.ox.ac.uk/files/7942/thesis.pdf.
[20] F. Croce, M. Lenzerini, A framework for explaining query answers in DL-Lite, in: C. Faron-Zucker, C. Ghidini, A. Napoli, Y. Toussaint (Eds.), Knowledge Engineering and Knowledge Management - 21st International Conference, EKAW 2018, volume 11313 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 83–97. doi:10.1007/978-3-030-03667-6_6.
[21] C. Alrabbaa, F. Baader, R. Dachselt, T. Flemisch, P. Koopmann, Visualising proofs and the modular structure of ontologies to support ontology repair, in: S. Borgwardt, T. Meyer (Eds.), Proceedings of the 33rd International Workshop on Description Logics (DL 2020), volume 2663 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2663/paper-2.pdf.
[22] A. Calì, G. Gottlob, T. Lukasiewicz, A general datalog-based framework for tractable query answering over ontologies, J. Web Semant. 14 (2012) 57–83. doi:10.1016/j.websem.2012.03.001.
[23] C. Alrabbaa, S. Borgwardt, P. Koopmann, A. Kovtunova, Finding good proofs for answers to conjunctive queries mediated by lightweight ontologies (technical report), 2022. URL: https://arxiv.org/abs/2206.09758. doi:10.48550/ARXIV.2206.09758.
[24] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning

and efficient query answering in description logics: The DL-Lite family, J. of Automated Reasoning 39 (2007) 385–429. doi:10.1007/s10817-007-9078-x.
[25] F. Baader, Terminological cycles in a description logic with existential restrictions, in: G. Gottlob, T. Walsh (Eds.), IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 2003, pp. 325–330. URL: http://ijcai.org/Proceedings/03/Papers/048.pdf.
[26] R. Fagin, P. G. Kolaitis, R. J. Miller, L. Popa, Data exchange: semantics and query answering, Theor. Comput. Sci. 336 (2005) 89–124. doi:10.1016/j.tcs.2004.10.033.
[27] R. Rosati, On conjunctive query answering in EL, in: D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, A. Turhan, S. Tessaris (Eds.), Proceedings of the 2007 International Workshop on Description Logics (DL2007), volume 250 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2007. URL: http://ceur-ws.org/Vol-250/paper_83.pdf.
[28] G. Gottlob, S. Kikot, R. Kontchakov, V. V. Podolskii, T. Schwentick, M. Zakharyaschev, The price of query rewriting in ontology-based data access, Artif. Intell. 213 (2014) 42–59. doi:10.1016/j.artint.2014.04.004.
[29] S. Borgwardt, W. Forkel, A. Kovtunova, Finding new diamonds: Temporal minimal-world query answering over sparse ABoxes, in: Proc. of the 3rd Int. Joint Conf. on Rules and Reasoning, RuleML+RR 2019, volume 11784 of *LNCS*, Springer, 2019, pp. 3–18. doi:10.1007/978-3-030-31095-0_1.
[30] S. Borgwardt, W. Forkel, A. Kovtunova, Temporal minimal-world query answering over sparse ABoxes, Theory Pract. Log. Program. 22 (2022) 193–228. URL: https://doi.org/10.1017/S1471068421000119. doi:10.1017/S1471068421000119.
[31] R. Alur, T. A. Henzinger, A really temporal logic, J. ACM 41 (1994) 181–204. doi:10.1145/174644.174651.
[32] V. Gutiérrez-Basulto, J. C. Jung, A. Ozaki, On metric temporal description logics, in: Proc. ECAI, IOS Press, 2016, pp. 837–845. doi:10.3233/978-1-61499-672-9-837.
[33] F. Baader, S. Borgwardt, P. Koopmann, A. Ozaki, V. Thost, Metric temporal description logics with interval-rigid names, in: Proc. of the 11th Int. Symp. on Frontiers of Combining Systems (FroCoS'17), Springer, 2017, pp. 60–76. doi:10.1007/978-3-319-66167-4_4.
[34] Y. Kazakov, M. Krötzsch, F. Simancik, The incredible ELK - From polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies, J. Autom. Reason. 53 (2014) 1–61. doi:10.1007/s10817-013-9296-3.