ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200153

Explanations for Ontology-Mediated Query Answering in Description Logics

İsmail İlkan Ceylan[1] and **Thomas Lukasiewicz**[2] and **Enrico Malizia**[3] and **Andrius Vaicenavičius**[4]

Abstract. Ontology-mediated query answering is a paradigm that seeks to exploit the semantic knowledge expressed in terms of ontologies to improve query answers over incomplete data sources. In this paper, we focus on description logic ontologies, and study the problem of *explaining* why an ontology-mediated query is *entailed* from a given data source. Specifically, we view explanations as minimal sets of assertions from an ABox, which satisfy the ontology-mediated query. Based on such explanations, we study a variety of problems taken from the recent literature on explanations (studied for existential rules), such as recognizing all minimal explanations. Our results establish tight connections between intractable explanation problems and variants of propositional satisfiability problems. We provide insights on the inherent computational difficulty of deriving explanations for ontology-mediated queries.

1 INTRODUCTION

Ontology-mediated query answering (OMQA) [35, 9] emerged as a sophisticated paradigm for accessing low-level data sources through a high-level logical theory (i.e., ontology). The idea is to encode background knowledge of an application domain in the form of an ontology, through which additional information (beyond the mere facts given in the data source) can be deduced. Extending the capabilities of data sources by allowing for a more complete set of answers to user queries, OMQA has become one of the focal points of research in knowledge representation and reasoning.

Description logics (DLs) [5] constitute a family of logic-based knowledge representation languages, which are commonly used to formulate ontologies. In the context of DLs, high-level domain knowledge is encoded into the TBox (i.e., ontology), and low-level, assertional knowledge is encoded into the ABox (i.e., data source). OMQA relative to the languages in the DL family is extensively studied, and it is well-understood. This theoretical understanding is supported by the various systems that have been developed to support OMQA in practice [44, 1].

Arguably, one of the most appealing aspects of *logical reasoning*is being able to *explain* the logical entailments. Having an unambiguous semantics, ontological knowledge can be interpreted, and its entailments can be explained relative to the axioms from the TBox and/or the assertions from the ABox. Explainability is an essential ingredient of various application domains, and so it has been widely investigated in DLs, though under different names such as *justifications*, *abduction*, and *axiom pinpointing*, as we outline next.

[1] University of Oxford, UK, email: ismail.ceylan@cs.ox.ac.uk
[2] University of Oxford, UK, and Alan Turing Institute, London, UK, email: thomas.lukasiewicz@cs.ox.ac.uk
[3] King's College London, UK, email: enrico.malizia@kcl.ac.uk
[4] University of Oxford, UK, email: andrius.vaicenavicius@cs.ox.ac.uk

The study of explanations in DLs dates back to [32, 10], where an explanation for a given subsumption is viewed as a *proof* with respect to the underlying proof calculi. A large of body of work provides explanations for subsumptions [25, 4, 41, 39], where an explanation is defined as a minimal subset of TBox axioms that suffices for the subsumption to be derived. This approach is sometimes called *axiom pinpointing* [4, 39, 34], as the task is to *pinpoint* a set of TBox *axioms* that are minimal and sufficient to entail the subsumption. The minimal set of axioms is commonly called *justifications* [25, 41]. Justifications are also essential for debugging unsatisfiable classes and contradictions; see, e.g., [38, 26] for explaining concept unsatisfiability, and [24] for explaining inconsistency. Various types of justifications were introduced in the literature depending on different notions of semantic minimality [23].

From a broader perspective, all these approaches can be viewed as a form of *logical abduction*, but all the works noted so far focus on the standard TBox reasoning tasks, such as subsumption checking and concept (un)satisfiability. Klarman et al. present a study based on ABox abduction [28], where the idea is to view explanations as sets of assertions, but differently from axiom pinpointing, the core aim is to explain *non-entailments* of instance queries. That is, given a non-entailment, the abduction task is to find a minimal set of (additional) facts such that, when the given ABox is extended to include these facts, the entailment holds [28]. An analogous study has been conducted by Calvanese et al. [14] for conjunctive queries, and for the *DL-Lite* family, where for a given a negative query answer (i.e., a non-entailment), the goal is to identify additions to the ABox that will ensure entailment. Despite the extensive research on explanations, the only work that provides explanations for (positive) ontology-mediated query answering in DLs is given for *DL-Lite* [11], where a polynomial-time algorithm for constructing explanations for ontology-mediated queries (OMQs) is presented.

In this work, we conduct a systematic study for explaining OMQA in order to close this gap. Note that a thorough study for explaining query entailments relative to existential rules has recently been given [16]: Briefly, given an OMQ, a minimal explanation, called MINEX, is defined as a minimal subset of facts from the database (or ABox) that entails the OMQ, while no strict subset of it does so. Based on this notion, several decision problems from the axiom pinpointing literature [34] are adapted to facilitate OMQA systems with explanations. These problems include the decision versions of identifying *a minimal explanation* (IS-MINEX), finding *all minimal explanations* (ALL-MINEX), finding *all explanations that contain some distinguished fact* (MINEX-REL), and finding *all explanations that do not contain a set of forbidden facts* (MINEX-IRREL).

We study these problems in the context of DLs towards providing a more complete picture of explanations for OMQA. For each expla-

Table 1: Syntax and semantics of concept descriptions (top), TBox axioms (middle), and ABox assertions (bottom) for DLs.

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
negation ($\mathcal{C}$)	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
union ($\mathcal{U}$)	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{a \mid \exists b \in C^{\mathcal{I}}.(a,b) \in r^{\mathcal{I}}\}$
universal restriction	$\forall r.C$	$\{a \mid \forall b \in C^{\mathcal{I}}.(a,b) \in r^{\mathcal{I}}\}$
nominal ($\mathcal{O}$)	$\{a\}$	$\{a^{\mathcal{I}}\}$
at least restrictions ($\mathcal{Q}$)	$\leq n\, r.C$	$\{d \mid \#\{e \mid (d,e) \in r^{\mathcal{I}}\} \leq n\}$
at most restrictions ($\mathcal{Q}$)	$\geq n\, r.C$	$\{d \mid \#\{e \mid (d,e) \in r^{\mathcal{I}}\} \geq n\}$
role inverse ($\mathcal{I}$)	r^-	$\{(b,a) \mid (a,b) \in r^{\mathcal{I}}\}$
	TBox Axioms	
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
role inclusion ($\mathcal{H}$)	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$
transitivity axiom	$trans(r)$	$r^{\mathcal{I}} = (r^{\mathcal{I}})^+$
	ABox Assertions	
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$r(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

nation problem, we conduct a detailed complexity analysis, relative to a wide range of OMQs. More specifically, we allow queries in the form of *instance queries*, or *unions of conjunctive queries*, which are coupled with ontologies formed in a DL that ranges from the light-weight *DL-Lite* and $\mathcal{EL}$ to expressive fragments such as $\mathcal{SHIQ}$. In our data complexity analysis, we show that *all* these problems are *tractable* for *DL-Lite* and *DL-Lite*$_{\mathcal{R}}$. Other tractability results in data complexity are given for $\mathcal{EL}$, $\mathcal{ELI}$, and Horn-$\mathcal{SHOIQ}$, for the problem of identifying minimal explanations. All the other results in data complexity confirm the *hardness* of deriving explanations, as we almost always observe an increase in the complexity in comparison to the complexity of OMQA. Our combined complexity analysis is less surprising in that the complexity is typically dominated by the complexity of OMQA, but in some cases (as we will highlight), we also observe an increase in the complexity. Detailed proofs are deferred to an extended version of this paper.

2 PRELIMINARIES

In this section, we give an overview of description logics, and ontology-mediated query answering, and provide the necessary computational complexity preliminaries relevant to our study.

2.1 Description logics

DLs are a family of knowledge representation languages, widely used in forming ontologies [18, 17]; they comprise prominent languages for ontology-mediated query answering.

Syntax. A *vocabulary* of a DL $\mathcal{L}$ consists of three countably infinite, pairwise disjoint sets of symbols: the set $\mathsf{N_I}$ of individual names, the set $\mathsf{N_C}$ of concept names, and the set $\mathsf{N_R}$ of role names. The building blocks of the DL syntax are individuals that represent concrete objects, concepts that represent sets of objects, and roles relating objects to objects via binary relations. DL concepts are built inductively from concept and role names, using the constructors given in Table 1.

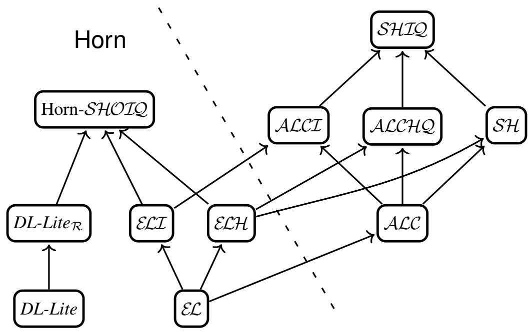

Figure 1: Semantic relationships between the DLs considered in this paper, where each arrow represents a language inclusion.

Knowledge base. Concepts are then used to form the axioms and assertions of the respective language, which are in the form of *general concept inclusions (GCIs)*, *role inclusions*, *transitivity axioms*, *concept assertions*, or *role assertions* as given in Table 1. A TBox axiom is a GCI, a role inclusion, or a transitivity axiom. An ABox assertion is either a concept assertion, or a role assertion as shown in Table 1. Finally, a TBox, or an ontology, is a finite set of axioms, an ABox is a finite set of assertions. Then, a knowledge base is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox, and $\mathcal{A}$ is an ABox.

Semantics. The semantics of DLs is given in terms of *interpretations*. An *interpretation* $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of a non-empty *interpretation domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$. This function assigns to each individual name $a \in \mathsf{N_I}$ an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, to each concept name $A \in \mathsf{N_C}$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to each role name $r \in \mathsf{N_R}$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Interpretations are then extended to arbitrary concept descriptions, axioms, and assertions as shown on the right-most column of Table 1.

An interpretation $\mathcal{I}$ is a *model of a TBox* $\mathcal{T}$, denoted $\mathcal{I} \models \mathcal{T}$, if it satisfies all the axioms in $\mathcal{T}$. Similarly, $\mathcal{I}$ is a *model of an ABox* $\mathcal{A}$, denoted $\mathcal{I} \models \mathcal{A}$, if it satisfies all the assertions in $\mathcal{T}$. Finally, $\mathcal{I}$ is a *model of a KB* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, denoted $\mathcal{I} \models \mathcal{K}$, if $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$.

Overview of the DL family. In the scope of this work, we consider a variety of different DLs that are all well-known in the literature. The DL $\mathcal{ALC}$ allows for *GCIs* as TBox axioms, and the concept language of $\mathcal{ALC}$ allows only for the constructors *top*, *bottom*, *negation*, *conjunction*, *disjunction*, *existential*, and *universal restrictions*. The DL $\mathcal{S}$ extends $\mathcal{ALC}$ with *transitivity axioms*. The letters $\mathcal{H}$, $\mathcal{I}$, $\mathcal{Q}$, and $\mathcal{O}$ denote *role inclusions*, *inverse roles*, *qualified number restrictions*, and *nominals*, respectively. Depending on what is allowed in the language, we obtain a variety of DLs, such as the very expressive DL $\mathcal{SHOIQ}$, which allows all the constructors given in Table 1.

The DL $\mathcal{EL}$ is a sub-logic of $\mathcal{ALC}$ which only allows the constructors *top*, *conjunction*, and *existential restrictions*. The DL *DL-Lite* allows axioms of the form $B \sqsubseteq C$, where B and C are concepts that are defined as follows:

$$B := A \mid \exists r.\top \mid \exists r^-.\top, \qquad C := B \mid \neg B.$$

The DL *DL-Lite*$_{\mathcal{R}}$ also allows a restricted type of role inclusions. For further details, we refer the reader to the relevant literature [5, 3]. Figure 1 summarizes the language inclusions between these DLs.

2.2 Ontology-mediated query answering

Ontology-mediated query answering generalizes the query answering task by incorporating knowledge in terms of an ontology. For-

Table 2: Complexity of OMQA relative to different DLs.

Description Logic	IQ		UCQ	
	data	comb.	data	comb.
DL-Lite, *DL-Lite*$_\mathcal{R}$	$\leq$ AC^0	NL	$\leq$ AC^0	NP
$\mathcal{EL}$, $\mathcal{ELH}$	P	P	P	NP
$\mathcal{ELI}$, Horn-$\mathcal{SHOIQ}$	P	EXP	P	EXP
$\mathcal{ALC}$, $\mathcal{ALCHQ}$	CONP	EXP	CONP	EXP
$\mathcal{ALCI}$, $\mathcal{SH}$, $\mathcal{SHIQ}$	CONP	EXP	CONP	2EXP

mally, an *ontology-mediated query* is a pair $(Q, \mathcal{T})$, where Q is a query, and $\mathcal{T}$ is an ontology. Given an ABox $\mathcal{A}$ and an OMQ $(Q, \mathcal{T})$, we say that $\mathcal{A}$ *entails* the OMQ $(Q, \mathcal{T})$, denoted $\mathcal{A} \models (Q, \mathcal{T})$, if for all models $\mathcal{I}$ of the knowledge base $(\mathcal{T}, \mathcal{A})$, we have that $\mathcal{I} \models Q$. *Ontology-mediated query answering* is the task of deciding whether $\mathcal{A} \models (Q, \mathcal{T})$ for a given ABox $\mathcal{A}$ and an OMQ $(Q, \mathcal{T})$.

We focus on two classes of queries, namely, *instance queries (IQs)* and *unions of conjunctive queries (UCQs)*. An *instance query (IQ)* is an assertion $C(a)$, where $C \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$. A *conjunctive query (CQ)* is a formula $\exists \mathbf{X}\, \Phi(\mathbf{X}, \mathbf{Y})$, where $\Phi(\mathbf{X}, \mathbf{Y})$ is a conjunction of expressions of the form $C(t_1)$ or $r(t_1, t_2)$, where $C \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and t_i is an individual or a variable. The variables in $\mathbf{X}$ are called *quantified*, and in $\mathbf{Y}$ *free*. A *union of conjunctive queries (UCQ)* is a disjunction of CQs. In this work, we focus on Boolean queries. Such queries have no free variables, that is, $\mathbf{Y} = \emptyset$.

We write $\mathrm{OMQA}(\mathcal{Q}, \mathcal{L})$ to refer to the problem of OMQA, where the query language $\mathcal{Q}$ can be instantiated either with IQs, or with UCQs, and $\mathcal{L}$ can be instantiated with a DL. We write $\mathcal{L}$-TBox to denote the class of TBoxes that can be formed in the DL $\mathcal{L}$.

2.3 Complexity background

In our complexity analysis, we make the standard assumptions [42]. Specifically, the *combined complexity* of OMQA is calculated by considering all the components as part of the input, i.e., the ABox, the TBox, and the query, and the *data complexity* is calculated by considering only the ABox as the input, i.e., everything else is assumed to be fixed.

The relations between the complexity classes that are most relevant for our analysis are given as follows:

$$\mathrm{AC}^0 \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP}, \mathrm{CONP} \subseteq \mathrm{D^P} \subseteq \Sigma_2^{\mathrm{P}}, \Pi_2^{\mathrm{P}} \subseteq \mathrm{EXP} \subseteq 2\mathrm{EXP}.$$

The complexity of query answering in DLs considered in this paper is presented in Table 2.

3 EXPLAINING QUERY ENTAILMENTS

We define the notion of a minimal explanation for a given OMQ relative to an ABox $\mathcal{A}$, as in earlier work [16]. Assuming the OMQ is entailed, we view minimal subsets of the ABox as explanations, provided that these sets are sufficient to entail the OMQ.

Definition 1 (MINEX)**.** Let $\mathcal{A}$ be an ABox and $(Q, \mathcal{T})$ be an OMQ such that $(\mathcal{T}, \mathcal{A})$ is consistent. An *explanation* for $(Q, \mathcal{T})$ in $\mathcal{A}$ is a subset $\mathcal{E} \subseteq \mathcal{A}$ such that $\mathcal{E} \models (Q, \mathcal{T})$.

A *minimal explanation* $\mathcal{E}$, or MINEX, for $(Q, \mathcal{T})$ in $\mathcal{A}$ is an explanation for $(Q, \mathcal{T})$ in $\mathcal{A}$ that is *subset-minimal for* $(Q, \mathcal{T})$, i.e., there is no proper subset $\mathcal{E}' \subsetneq \mathcal{E}$ that is an explanation for $(Q, \mathcal{T})$ in $\mathcal{A}$.

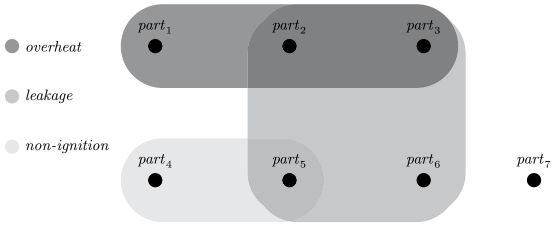

Figure 2: Illustration of possible faults in different parts of a system.

We say that $\mathcal{E}$ is a MINEX for $(Q, \mathcal{T})$ if the ABox $\mathcal{A}$ is clear from the context. Importantly, this definition assumes the initial knowledge base to be consistent, which is not explicit in the definition of earlier work [16], but it is implied, since that work focuses on existential rules without any negative constraints.

The choice of consistency of the initial knowledge base helps us to identify the complexity of recognizing explanations more precisely. More specifically, as $(\mathcal{T}, \mathcal{A})$ is assumed to be consistent, for any subset $\mathcal{B} \subseteq \mathcal{A}$, the knowledge base $(\mathcal{T}, \mathcal{B})$ is also consistent. This holds, as every model of $(\mathcal{T}, \mathcal{A})$ is also a model of $(\mathcal{T}, \mathcal{B})$. In particular, $(\mathcal{T}, \mathcal{E})$ is consistent for any candidate MINEX $\mathcal{E}$ for $(Q, \mathcal{T})$ in $\mathcal{A}$.

MINEXS are tightly connected to *minimal hitting sets*, and therefore can be applied in a variety of fields [21], including, computational biology [16, 27], data mining [13], and model-based fault diagnosis [36]. Our running example is a *fault diagnosis* problem.

Example 2. Suppose we have an engine that might experience a number of possible failures, each caused by a fault of one of the constituent parts, presented in Figure 2. We can encode *fault diagnosis* as follows. We define an ABox $\mathcal{A} = \mathcal{A}_{faults} \cup \mathcal{A}_{st}$, where the first component of the ABox encodes the parts that may fail:

$$\mathcal{A}_{faults} = \{Fault(part_i) \mid 1 \leq i \leq 7\}.$$

The second component of the ABox encodes causes of failures:

$$\begin{aligned} \mathcal{A}_{st} = \; & \{caused(overheat, part_i) \mid i \in \{1, 2, 3\}\} \cup \\ & \{caused(leakage, part_i) \mid i \in \{2, 3, 5, 6\}\} \cup \\ & \{caused(non\text{-}ignition, part_i) \mid i \in \{4, 5\}\}. \end{aligned}$$

The TBox consists of a single axiom that expresses that, if a part failed, it causes the corresponding failures:

$$\mathcal{T} = \{\exists caused.Fault \sqsubseteq Failure\}.$$

The query asks for a minimal set of faults that explains the observed failures together with the structural facts of the engine:

$$\begin{aligned} Q = \; & Failure(overheat) \wedge Failure(leakage) \wedge \\ & Failure(non\text{-}ignition) \wedge \bigwedge_{\psi \in \mathcal{A}_{st}} \psi. \end{aligned}$$

Any MINEX for $(Q, \mathcal{T})$ in $\mathcal{A}$ is a minimal subset of *Fault* concepts, together with all assertions in $\mathcal{A}_{st}$, that lead to *overheat*, *leakage*, and *non-ignition*. In particular, MINEXS for $(Q, \mathcal{T})$ in $\mathcal{A}$ correspond to minimal covers of the graph in Figure 2. ■

4 DECISION PROBLEMS FOR EXPLANATIONS

In the following, we introduce the key decision problems for our study. We investigate four different problems, namely, recognizing whether a set is a MINEX, recognizing if a set contains all MINEXS, deciding if a fact is contained in some MINEX, and deciding if there is a MINEX that does not contain any forbidden set.

4.1 A minimal explanation

A fundamental problem related to minimal explanations is the problem of recognizing a MINEX.

Problem: IS-MINEX$(\mathcal{Q},\mathcal{L})$
Input: An ABox $\mathcal{A}$, an OMQ $(Q,\mathcal{T})$, and a subset $\mathcal{E}\subseteq\mathcal{A}$, where $(\mathcal{T},\mathcal{A})$ is consistent, Q is a query from the class $\mathcal{Q}$, and $\mathcal{T}$ is an $\mathcal{L}$-TBox.
Question: Is $\mathcal{E}$ a MINEX for $(Q,\mathcal{T})$ in $\mathcal{A}$?

Let us illustrate IS-MINEX on our running example.

Example 3. Consider the ABox $\mathcal{A}$ from the running example, and the following subsets of it:

$$\mathcal{E}_1=\{\mathit{Fault}(\mathit{part}_2),\mathit{Fault}(\mathit{part}_5\}\cup\mathcal{A}_{st},$$
$$\mathcal{E}_2=\{\mathit{Fault}(\mathit{part}_1),\mathit{Fault}(\mathit{part}_4),\mathit{Fault}(\mathit{part}_6)\}\cup\mathcal{A}_{st},$$
$$\mathcal{E}_3=\{\mathit{Fault}(\mathit{part}_1),\mathit{Fault}(\mathit{part}_4),\mathit{Fault}(\mathit{part}_5)\}\cup\mathcal{A}_{st},\ \text{and}$$
$$\mathcal{E}_4=\{\mathit{Fault}(\mathit{part}_4),\mathit{Fault}(\mathit{part}_6)\}\cup\mathcal{A}_{st}.$$

Note that $\mathcal{E}_1$ and $\mathcal{E}_2$ are MINEXS for $(Q,\mathcal{T})$ in $\mathcal{A}$, while $\mathcal{E}_3$ and $\mathcal{E}_4$ are not. $\mathcal{E}_3$ is not a MINEX for $(Q,\mathcal{T})$ in $\mathcal{A}$, since its subset $\{\mathit{Fault}(\mathit{part}_1),\mathit{Fault}(\mathit{part}_5)\}\cup\mathcal{A}_{st}$ entails $(Q,\mathcal{T})$, and so $\mathcal{E}_3$ is not minimal. $\mathcal{E}_4$ is not a MINEX for $(Q,\mathcal{T})$ in $\mathcal{A}$, as $\mathcal{E}_4\not\models(Q,\mathcal{T})$. ■

The generic approach for solving IS-MINEX [16] also applies here: To ensure that a set $\mathcal{E}$ is a MINEX for the OMQ, we need to check that it entails the query (which can be done in $\mathcal{C}$), and that it is minimal. The latter can be done by checking whether removing any single assertion from $\mathcal{E}$ would invalidate the OMQ, and hence with $|\mathcal{E}|$ number of co-$\mathcal{C}$ checks. So, we state the following corollary to [16, Theorem 4].

Corollary 4. IS-MINEX$(\mathcal{Q},\mathcal{L})$ *can be decided by a single $\mathcal{C}$ check, and a linear number (in the size of data) of co-$\mathcal{C}$ checks, where $\mathcal{C}$ is the complexity of* OMQA$(\mathcal{Q},\mathcal{L})$.

Corollary 4 covers all the membership results shown in Table 3. For example, if $\mathcal{C}=\mathrm{AC}^0$, we need a single AC^0 and a linear number of co-AC^0 checks to solve IS-MINEX$(\mathcal{Q},\mathcal{L})$. Since AC^0 is closed under complement, we have that a linear number of AC^0 checks is sufficient. They can be combined with an AND gate to obtain an AC^0 circuit (i.e., fixed depth and polynomial size) that solves IS-MINEX$(\mathcal{Q},\mathcal{L})$.

If, on the other hand, $\mathcal{C}=$ CONP, we can encode a linear number of CONP checks as a single CONP check. To see that, note that we can solve the complement problem in NP, as we can combine linearly many NP-certificates as a single NP-certificate of polynomial size in the size of the input. Therefore, we can conclude that IS-MINEX$(\mathcal{Q},\mathcal{L})$ is in $\mathrm{D}^{\mathrm{P}}=\mathrm{NP}\wedge$ CONP in this case. Other membership results follow by analogous arguments.

We proceed with the hardness results, presented in Table 3, which are novel. First, we show that OMQA$(\mathrm{IQ},\mathcal{ELI})$ can be reduced to IS-MINEX$(\mathrm{IQ},\mathcal{ELI})$ in data complexity, and thus recognizing a MINEX is not easier than OMQA. In this reduction, we replace each role assertion by two role assertions and introduce reachability type axioms to the TBox to force each MINEX to contain all assertions from the constructed ABox. Thus, an ABox entails an OMQ iff the extended ABox is a MINEX for the OMQ. The combined complexity results borrow ideas from this data complexity reduction.

Table 3: Complexity results for IS-MINEX$(\mathcal{Q},\mathcal{L})$.

Description Logic	IQ		UCQ	
	data	comb.	data	comb.
DL-Lite, *DL-Lite*$_\mathcal{R}$	$\leq\mathrm{AC}^0$	$\leq$ P	$\leq\mathrm{AC}^0$	D^{P}
$\mathcal{EL},\mathcal{ELH}$	$\leq$ P	P	$\leq$ P	D^{P}
$\mathcal{ELI}$, Horn-$\mathcal{SHOIQ}$	P	EXP	P	EXP
$\mathcal{ALC},\mathcal{ALCHQ}$	D^{P}	EXP	D^{P}	EXP
$\mathcal{ALCI},\mathcal{SH},\mathcal{SHIQ}$	D^{P}	EXP	D^{P}	2EXP

Theorem 5. IS-MINEX$(\mathrm{IQ},\mathcal{ELI})$ *is* P*-hard in data complexity, and* EXP*-hard in combined complexity.* IS-MINEX$(\mathrm{UCQ},\mathcal{ALCI})$ *and* IS-MINEX$(\mathrm{UCQ},\mathcal{SH})$ *are* 2EXP*-hard in combined complexity.*

It is unclear whether IS-MINEX$(\mathrm{IQ},\mathcal{EL})$ is also P-hard in data complexity, so it is open whether Theorem 5 can be strengthened in this direction. This is so, because our proof heavily relies on inverse roles to enforce all ABox assertions to be included in the minimal explanation, which is necessary for IS-MINEX to simulate OMQA. It is unclear whether a similar simulation can be obtained in $\mathcal{EL}$.

EXP-hardness of IS-MINEX(IQ, $\mathcal{ELI}$) in combined complexity is immediate given the data complexity reduction, and implies lower bounds for (IQ, Horn-$\mathcal{SHOIQ}$), (IQ, $\mathcal{ALCI}$), and (IQ, $\mathcal{SHIQ}$) due to language inclusions. 2EXP-hardness of IS-MINEX(UCQ, $\mathcal{ALCI}$) and hence of IS-MINEX$(\mathrm{UCQ},\mathcal{SHIQ})$ follow by a slightly modified proof. The proof for IS-MINEX$(\mathrm{UCQ},\mathcal{SH})$ requires a further modification due to the lack of inverse roles in $\mathcal{SH}$, but we make use of the UCQ and the role inclusions to obtain the result.

All the remaining P- and EXP-hardness results from Table 3 are covered by the following result.

Theorem 6. IS-MINEX$(\mathrm{IQ},\mathcal{EL})$ *is* P*-hard;* IS-MINEX$(\mathrm{IQ},\mathcal{ELU})$ *is* EXP*-hard in combined complexity.*

We reduce from subsumption checking in $\mathcal{EL}$ and $\mathcal{ELU}$ to IS-MINEX$(\mathrm{IQ},\mathcal{EL})$ and IS-MINEX$(\mathrm{IQ},\mathcal{ELU})$, respectively. Theorem 6 also implies lower bounds for IS-MINEX$(\mathrm{IQ},\mathcal{ALC})$.

Our next result covers all D^{P}-hardness results for IS-MINEX in data complexity, and is arguably the most interesting result of this section. This result is a consequence of CONP-hardness of OMQA in $\mathcal{ALC}$ in data complexity, and therefore differs from the data complexity results obtained for existential rules.

Theorem 7. IS-MINEX$(\mathrm{IQ},\mathcal{ALC})$ *is* D^{P}*-hard in data complexity.*

Proof sketch. The reduction is from the D^{P}-complete problem IS-MUS [33] to IS-MINEX$(\mathrm{IQ},\mathcal{ALC})$ in data complexity. The problem IS-MUS asks if a 3CNF ϕ is a minimal unsatisfiable formula (MUS), that is, ϕ is unsatisfiable, but removing any clause renders ϕ satisfiable. Note that IS-MUS is close in nature to IS-MINEX.

We encode ϕ in the ABox $\mathcal{A}_\phi$ as follows. For each clause c_j with i negative literals, we add a fact $C_i(c_j)$ to $\mathcal{A}_\phi$, and, for each $i\in\{1,2,3\}$, add a fact $r_{p_i}(c_j,x_k)$ or $r_{n_i}(c_j,x_k)$ to encode that x_k appears as a positive or a negative literal in c_j in the ith position, respectively. We further add assertions $\mathit{Tau}(\tau)$ and $r_0(\tau,x_k)$ for each variable x_k in ϕ, which are needed to establish the correspondence between minimal unsatisfiability and minimal explanations.

The TBox $\mathcal{T}_\phi$ captures the satisfiability of ϕ. It asserts that in each clause c_j at least one literal has to be satisfied. It also ensures that if some variable is assigned both values *true* and *false*, then $\mathit{Unsat}(\tau)$ holds. We take the query to be $\mathit{Unsat}(\tau)$. Then, $\mathit{Unsat}(\tau)$ is entailed if there is no satisfying assignment to ϕ. This construction ensures that $\mathcal{A}_\phi$ is a MINEX for $(\mathit{Unsat}(\tau),\mathcal{T}_\phi)$ iff ϕ is a MUS. □

Table 4: Complexity results for ALL-MINEX$(\mathcal{Q}, \mathcal{L})$.

Description Logic	IQ		UCQ	
	data	comb.	data	comb.
DL-Lite, *DL-Lite*$_\mathcal{R}$	$\leq$ P	$\leq$ P	$\leq$ P	$\mathrm{D^P}$
$\mathcal{EL}, \mathcal{ELH}$	coNP	coNP	coNP	$\mathrm{D^P}$
$\mathcal{ELI}$, Horn-$\mathcal{SHOIQ}$	coNP	EXP	coNP	EXP
$\mathcal{ALC}, \mathcal{ALCHQ}$	Π_2^P	EXP	Π_2^P	EXP
$\mathcal{ALCI}, \mathcal{SH}, \mathcal{SHIQ}$	Π_2^P	EXP	Π_2^P	2EXP

It remains to show the $\mathrm{D^P}$-hardness results in combined complexity for light-weight DLs. We reduce from 3COL-NONCOL of determining whether one graph is 3-colourable and another not 3-colourable to IS-MINEX$(\mathrm{UCQ}, \mathcal{L})$. We encode the two graphs into the query and the allowed colourings of the edges into the ABox. Our result applies to any DL, as it is shown relative to an empty TBox.

Theorem 8. IS-MINEX$(\mathrm{UCQ}, \mathcal{L})$ *is* $\mathrm{D^P}$*-hard in combined complexity for any DL* $\mathcal{L}$*.*

This concludes the results given in Table 3, and our analysis for IS-MINEX.

4.2 All minimal explanations

Another important problem related to MINEXS is the problem of recognizing the set of all MINEXS.

Problem: ALL-MINEX$(\mathcal{Q}, \mathcal{L})$
Input: An ABox $\mathcal{A}$, an OMQ $(Q, \mathcal{T})$, and a set $\mathfrak{E} \subseteq \mathcal{P}(\mathcal{A})$ of subsets of $\mathcal{A}$, where the query Q is from the class $\mathcal{Q}$, and $\mathcal{T}$ is an $\mathcal{L}$-TBox.
Question: Is $\mathfrak{E}$ precisely the set of all MINEXS for $(Q, \mathcal{T})$ in $\mathcal{A}$?

Example 9. In our example, the set $\mathfrak{E}$ of all MINEXS for $(Q, \mathcal{T})$ in the ABox $\mathcal{A}$ is

$$\begin{aligned}\mathfrak{E} = \big\{&\{Fault(part_1), Fault(part_4), Fault(part_6)\} \cup \mathcal{A}_{st},\\ &\{Fault(part_1), Fault(part_5)\} \cup \mathcal{A}_{st},\\ &\{Fault(part_2), Fault(part_4)\} \cup \mathcal{A}_{st},\\ &\{Fault(part_2), Fault(part_5)\} \cup \mathcal{A}_{st},\\ &\{Fault(part_3), Fault(part_4)\} \cup \mathcal{A}_{st},\\ &\{Fault(part_3), Fault(part_5)\} \cup \mathcal{A}_{st}\big\}.\end{aligned}$$

Notice that there is no minimal explanation outside of this set. ■

To decide whether $\mathfrak{E}$ contains all the MINEXS, we check whether each element in $\mathfrak{E}$ entails the OMQ with a linear number of $\mathcal{C}$ checks. Then, observe that each set in $\mathfrak{E}$ is minimal, and there is no MINEX outside of $\mathfrak{E}$, iff $\mathcal{E}' \not\models (Q, \mathcal{T})$ for each $\mathcal{E}' \subseteq \mathcal{A}$ such that $\mathcal{E} \not\subseteq \mathcal{E}'$ for any $\mathcal{E} \in \mathfrak{E}$. To ensure this, we need a co-$(\mathrm{NP}^\mathcal{C})$ check. Thus, we can again give a general membership result for ALL-MINEX as a corollary to a result on existential rules [16, Theorem 8].

Corollary 10. ALL-MINEX$(\mathcal{Q}, \mathcal{L})$ *can be decided by a linear number of* $\mathcal{C}$ *checks and a single co-*$(\mathrm{NP}^\mathcal{C})$ *check, where* $\mathcal{C}$ *is the complexity of* OMQA$(\mathcal{Q}, \mathcal{L})$*.*

This implies all but the P-membership results in Table 4. If the OMQ is FO-rewritable, then all explanations can be computed in polynomial time in data complexity. The following result is a corollary to [16, Theorem 9] and [7, Proposition 4.8].

Corollary 11. *For any DL-Lite*$_\mathcal{R}$*-TBox* $\mathcal{T}$*, any ABox* $\mathcal{A}$*, and any UCQ* Q*, computing all* MINEXS *for an OMQ* $(Q, \mathcal{T})$ *in* $\mathcal{A}$ *is feasible in polynomial time in data complexity.*

The above implies that ALL-MINEX is in P in data complexity when the query is IQ or UCQ and the TBox is a *DL-Lite* or *DL-Lite*$_\mathcal{R}$ TBox. This result can be generalized to any FO-rewritable language.

The only missing membership results for ALL-MINEX are for (IQ, *DL-Lite*) and (IQ, *DL-Lite*$_\mathcal{R}$) in combined complexity. Importantly, *DL-Lite*$_\mathcal{R}$ does not allow conjunctions on the left-hand side of the TBox axioms, and any concept has *at most one* role or concept name. Hence, every MINEX for *DL-Lite*$_\mathcal{R}$ can only be of size 1 for instance queries. This means that we can go through each ABox assertion and see whether each is a MINEX in polynomial time. This implies that the check whether a given candidate set $\mathfrak{E}$ is exactly the set of all MINEXS is feasible in polynomial time.

Lemma 12. *For any DL-Lite*$_\mathcal{R}$*-TBox* $\mathcal{T}$*, any ABox* $\mathcal{A}$*, and any instance query* Q*, computing the set of all* MINEXS *for* $(Q, \mathcal{T})$ *in* $\mathcal{A}$ *is in polynomial time in combined complexity.*

We proceed with the hardness results, presented in Table 4. We show that ALL-MINEX$(\mathrm{IQ}, \mathcal{EL})$ is coNP-hard in data complexity. We show it by reducing a coNP-complete problem that has a very close nature to ALL-MINEX, namely, ALL-MV [6, Lemma 1]. This problem asks to decide if a given set $\mathfrak{V}$ contains *all* minimal valuations satisfying a Boolean monotone formula ϕ. In our reduction, the ABox $\mathcal{A}_\phi$ encodes ϕ, and $\mathcal{T}$ is empty. Each MINEX for $(Q, \mathcal{T})$ in $\mathcal{A}_\phi$ corresponds to a minimal valuation for ϕ. This result implies all the coNP-hardness results for ALL-MINEX, presented in Table 4.

Theorem 13. ALL-MINEX$(\mathrm{IQ}, \mathcal{EL})$ *is* coNP*-hard in data complexity.*

Note that this result strengthens the result given in [16, Theorem 9], since $\mathcal{EL}$ can be embedded in guarded existential rules.

We now show Π_2^P-hardness of ALL-MINEX$(\mathrm{IQ}, \mathcal{ALC})$ in data complexity, which is the most intricate result of this section and builds on the ideas presented in the proof of Theorem 7.

Theorem 14. ALL-MINEX$(\mathrm{IQ}, \mathcal{ALC})$ *is* Π_2^P*-hard in data complexity.*

Proof sketch. We give a reduction from the Π_2^P-complete problem $\forall\exists$3SAT [40, 43], which asks, whether $\forall X\, \exists Y\, \phi(X, Y)$ is satisfiable, where $\phi(X, Y)$ is a 3CNF formula. We encode $\phi(X, Y)$ in the ABox $\mathcal{A}_\phi$ in the same way as in the proof of Theorem 7: we just add the facts r_{p_i} and r_{n_i} both for variables in X and Y. Furthermore, we add two assertions $T(x_j)$ and $F(x_j)$ to $\mathcal{A}_\phi$ for each variable x_j in X. They correspond to x_j being assigned *true* and *false*. We take the TBox $\mathcal{T}$ and the query $Q = \mathit{Unsat}(\tau)$ as in the proof of Theorem 7.

Then, $\mathfrak{E} = \{\{\mathit{Tau}(\tau), r_0(\tau, x_j), T(x_j), F(x_j)\} \mid x_j \text{ a var. in } \phi\}$ is the set of all MINEXS for $(Q, \mathcal{T})$ in $\mathcal{A}_\phi$ iff $\forall X\, \exists Y\, \phi(X, Y)$ is satisfiable. To see that this claim holds, first note that each set $\mathcal{E}$ in $\mathfrak{E}$ is a MINEX for $(Q, \mathcal{T})$, as $\mathcal{E}$ contains the four facts needed for the axiom $\mathit{Tau} \sqcap \exists r_0.(T \sqcap F) \sqsubseteq \mathit{Unsat}$ to force $\mathit{Unsat}(\tau)$ to hold in every model. Note also that, as in the proof of Theorem 7, if $\forall X\, \exists Y\, \phi(X, Y)$ is satisfiable, then there are no other MINEXS outside $\mathfrak{E}$. Additionally, if $\forall X\, \exists Y\, \phi(X, Y)$ was not satisfiable, then there would be a truth assignment v_X to variables in X such that, for any choice of values v_Y for Y, $\phi(v_X(X), v_Y(Y))$ is false. In this case, some subset of $\{\mathit{Tau}(\tau), r_0(\tau, x_j) \mid x_j \text{ a var. in } \phi\} \cup \{T(x_j) \mid v_X(x_j) = T\} \cup \{F(x_j) \mid v_X(x_j) = F\}$ would be a MINEX, not contained in $\mathfrak{E}$. □

Table 5: Complexity results for MINEX-IRREL$(\mathcal{Q}, \mathcal{L})$.

Description Logic	IQ		UCQ	
	data	comb.	data	comb.
DL-Lite, *DL-Lite*$_{\mathcal{R}}$	$\leq$ P	$\leq$ P	$\leq$ P	NP
$\mathcal{EL}, \mathcal{ELH}$	NP	NP	NP	NP
$\mathcal{ELI}$, Horn-$\mathcal{SHOIQ}$	NP	EXP	NP	EXP
$\mathcal{ALC}, \mathcal{ALCHQ}$	Σ_2^{P}	EXP	Σ_2^{P}	EXP
$\mathcal{ALCI}, \mathcal{SH}, \mathcal{SHIQ}$	Σ_2^{P}	EXP	Σ_2^{P}	2EXP

The remaining hardness results, such as EXP- or 2EXP-hardness, follow immediately from the proofs of the IS-MINEX results. This concludes our analysis for ALL-MINEX.

4.3 Explanations avoiding forbidden sets

In some applications, we are interested in the existence of a MINEX avoiding certain configurations, that is, the problem of deciding if there is a MINEX that does not contain any of the forbidden sets of assertions.

Problem: MINEX-IRREL$(\mathcal{Q}, \mathcal{L})$
Input: An ABox $\mathcal{A}$, an OMQ $(Q, \mathcal{T})$, and a set $\mathfrak{F} \subseteq \mathcal{P}(\mathcal{A})$ of subsets of $\mathcal{A}$, where the query Q is from the class of $\mathcal{Q}$, and $\mathcal{T}$ is an $\mathcal{L}$-TBox.
Question: Is there a MINEX $\mathcal{E}$ for $(Q, \mathcal{T})$ in $\mathcal{A}$ such that $\mathcal{F} \not\subseteq \mathcal{E}$ for every $\mathcal{F} \in \mathfrak{F}$?

We illustrate MINEX-IRREL on our running example, by assuming that in our engine, some sets of parts cannot *all* be faulty. This constraint can be formulated as an instance of MINEX-IRREL.

Example 15. Let the following set

$$\mathfrak{F}_1 = \big\{\{\mathit{Fault}(\mathit{part}_5)\}, \{\mathit{Fault}(\mathit{part}_1), \mathit{Fault}(\mathit{part}_6)\}\big\}$$

be the set of forbidden sets. Then, observe, for example, that the set $\{\mathit{Fault}(\mathit{part}_2), \mathit{Fault}(\mathit{part}_4)\} \cup \mathcal{A}_{st}$ is a MINEX for $(Q, \mathcal{T})$ in $\mathcal{A}$ that does not contain any of the sets in $\mathfrak{F}_1$. On the other hand, if the forbidden sets are $\mathfrak{F}_2 = \mathfrak{F}_1 \cup \big\{\{\mathit{Fault}(\mathit{part}_4)\}\big\}$, then there is no MINEX for $(Q, \mathcal{T})$ in $\mathcal{A}$ such that no set in $\mathfrak{F}_2$ is contained in $\mathcal{E}$. ■

As before, we show a generic membership result that follows from [16, Theorem 13]. MINEX-IRREL can be solved by first guessing a set $\mathcal{E}$ not containing any of the forbidden sets in NP, and then checking in $\mathcal{C}$, whether $\mathcal{E}$ entails the OMQ. If $\mathcal{E}$ entails the OMQ, then it must contain a MINEX as a subset (as a simple consequence of the monotonicity of the entailment relation).

Corollary 16. MINEX-IRREL$(\mathcal{Q}, \mathcal{L})$ *can be decided in* NP*, calling a check in* $\mathcal{C}$*, where* $\mathcal{C}$ *is the complexity of* OMQA$(\mathcal{Q}, \mathcal{L})$.

Corollary 16 covers all membership results, presented in Table 5, except for the P-membership results. If $\mathcal{C}$ = NP, then two NP certificates can be combined as a single NP certificate, and thus MINEX-IRREL is in NP. If, for example, $\mathcal{C}$ = CONP, then MINEX-IRREL is in Σ_2^{P}.

The upper bounds for *DL-Lite*$_{\mathcal{R}}$ is due to similar reasons as before, since we can construct the set of *all* MINEXS in polynomial time by Corollary 11, and then check in polynomial time if this set contains a MINEX that does not contain any forbidden set.

Corollary 17. *For any DL-Lite*$_{\mathcal{R}}$*-TBox* $\mathcal{T}$*, any ABox* $\mathcal{A}$*, and any UCQ* Q*, finding a* MINEX *for an OMQ* $(Q, \mathcal{T})$ *in a ABox* $\mathcal{A}$ *that does not contain forbidden set of assertions* $\mathfrak{F}$ *is feasible in polynomial time in data complexity.*

Table 6: Complexity results for MINEX-REL$(\mathcal{Q}, \mathcal{L})$.

Description Logic	IQ		UCQ	
	data	comb.	data	comb.
DL-Lite, *DL-Lite*$_{\mathcal{R}}$	$\leq$ P	$\leq$ P	$\leq$ P	Σ_2^{P}
$\mathcal{EL}, \mathcal{ELH}$	NP	NP	NP	Σ_2^{P}
$\mathcal{ELI}$, Horn-$\mathcal{SHOIQ}$	NP	EXP	NP	EXP
$\mathcal{ALC}, \mathcal{ALCHQ}$	Σ_2^{P}	EXP	Σ_2^{P}	EXP
$\mathcal{ALCI}, \mathcal{SH}, \mathcal{SHIQ}$	Σ_2^{P}	EXP	Σ_2^{P}	2EXP

This proves that MINEX-IRREL for (UCQ, *DL-Lite*) and (UCQ, *DL-Lite*$_{\mathcal{R}}$) are in P in data complexity. Our arguments here are of a general nature in the sense that they can be extended to any FO-rewritable OMQ.

We proceed with the hardness results for MINEX-IRREL. First, we show the NP-hardness for MINEX-IRREL(IQ, $\mathcal{EL}$) in data complexity, as an adaptation of the result presented in [16, Theorem 15] to the DL $\mathcal{EL}$ and to instance queries. This reduction is from the NP-complete problem PATH WITH FORBIDDEN PAIRS [20, 22].

Theorem 18. MINEX-IRREL(IQ, $\mathcal{EL}$) *is* NP*-hard in data complexity.*

It remains to show that MINEX-IRREL(IQ, $\mathcal{ALC}$) is Σ_2^{P}-hard in data complexity, which implies all Σ_2^{P}-hardness results in Table 5.

Theorem 19. MINEX-IRREL(IQ, $\mathcal{ALC}$) *is* Σ_2^{P}*-hard in data complexity.*

Proof sketch. We provide a reduction from a Σ_2^{P}-complete problem $\mathrm{QBF}_{2,\forall,\neg}^{CNF}$ [37] of determining whether a formula $\exists X \forall Y \neg\phi(X, Y)$, where ϕ is 3CNF, is valid. The construction is an extension of the proof of Theorem 14. In particular, given a formula $\exists X \forall Y \neg\phi(X, Y)$, the ABox $\mathcal{A}_\phi$ encodes $\phi(X, Y)$ and both the T and F assignment to each variable in X. The forbidden sets are of the form $\{T(x_i), F(x_i)\}$, where $x_i \in X$. The TBox ensures that the instance query is entailed iff there is no assignment to Y that entails the query. □

Finally, if we do not forbid any of the sets of the ABox assertions, then MINEX-IRREL is equivalent to OMQA. This observation implies that the remaining hardness results in Table 5 follow from the hardness of OMQA.

4.4 Explanations containing a distinguished fact

The last problem that we investigate is related to the role of an individual assertion in explaining OMQA. This problem asks whether there is a MINEX containing a distinguished assertion.

Problem: MINEX-REL$(\mathcal{Q}, \mathcal{L})$
Input: An ABox $\mathcal{A}$, an OMQ $(Q, \mathcal{T})$, and an assertion $\psi \in \mathcal{A}$, where the query Q is from the class $\mathcal{Q}$, and $\mathcal{T}$ is a $\mathcal{L}$-TBox.
Question: Is there a MINEX $\mathcal{E}$ for $(Q, \mathcal{T})$ in $\mathcal{A}$ such that $\psi \in \mathcal{E}$?

Example 20. Suppose that we want to know whether a particular part can be contributing to the observed failures. That is, we want to find out whether $\psi_1 = \mathit{Fault}(\mathit{part}_1)$ or $\psi_2 = \mathit{Fault}(\mathit{part}_7)$ can be part of an explanation for the observed failures. Note that there is a MINEX for $(Q, \mathcal{T})$ in $\mathcal{A}$ that contains ψ_1, for example, $\{\mathit{Fault}(\mathit{part}_1), \mathit{Fault}(\mathit{part}_5)\} \cup \mathcal{A}_{st}$. On the other hand, there is no MINEX for $(Q, \mathcal{T})$ in $\mathcal{A}$ that contains ψ_2. ■

In order to solve MINEX-REL, we can guess a subset of the ABox that contains a distinguished fact in NP, and then check if it is a MINEX using an oracle for IS-MINEX, as in [16, Theorem 17], which yields the following corollary.

Corollary 21. MINEX-REL$(\mathcal{Q}, \mathcal{L})$ *is in* NP*, calling a single check in* IS-MINEX$(\mathcal{Q}, \mathcal{L})$.

This result covers all membership results, presented in Table 6, apart from the membership in P. In particular, if IS-MINEX$(\mathcal{Q}, \mathcal{L})$ is in $\mathrm{D^P}$, then, since $\mathrm{D^P} = \mathrm{NP} \wedge \mathrm{coNP}$, MINEX-REL$(\mathcal{Q}, \mathcal{L})$ can be decided in $\mathrm{NP^{coNP}} = \Sigma_2^{\mathrm{P}}$.

To solve MINEX-REL for FO-rewritable languages, we can first compute the set of all MINEXS in polynomial time (Corollary 11) and then check if any of these MINEXS contains a distinguished assertion, which yields the following result.

Corollary 22. *For any DL-Lite$_\mathcal{R}$-TBox $\mathcal{T}$, any ABox $\mathcal{A}$, and any UCQ Q, finding a* MINEX *for an OMQ $(Q, \mathcal{T})$ in a ABox $\mathcal{A}$ over $\mathcal{L}$ that contains a distinguished assertion ψ is feasible in polynomial time in data complexity.*

Our next result concerns MINEX-REL$(\mathrm{IQ}, \mathcal{EL})$, and is an adaptation of [16, Theorem 19] to the DL $\mathcal{EL}$ and instance queries: we reduce from the NP-complete problem PATH VIA NODE [29].

Theorem 23. MINEX-REL$(\mathrm{IQ}, \mathcal{EL})$ *is* NP*-hard in data complexity.*

The most interesting result of this section is also related to the DL $\mathcal{ALC}$. We show that MINEX-REL$(\mathrm{IQ}, \mathcal{ALC})$ is Σ_2^{P}-hard in data complexity. This implies all Σ_2^{P}-hardness results in data complexity in Table 6. The proof of the following theorem borrows ideas from the proof of Theorem 19

Theorem 24. MINEX-REL$(\mathrm{IQ}, \mathcal{ALC})$ *is Σ_2^{P}-hard in data complexity.*

Finally, we show the Σ_2^{P}-hardness results in combined complexity, where the complexity is one level higher up in the polynomial hierarchy, as compared to the complexity of OMQA in the respective languages. We reduce the Σ_2^{P}-hard problem $\mathrm{QBF}_{2,\forall,\neg}^{CNF}$ to MINEX-REL$(\mathrm{UCQ}, \mathcal{L})$ in combined complexity, where $\mathcal{L}$ may be any DL, as we take the TBox to be empty in the reduction.

Theorem 25. MINEX-REL$(\mathrm{UCQ}, \mathcal{L})$ *is Σ_2^{P}-hard in combined complexity for any DL $\mathcal{L}$.*

The remaining hardness results follow from the fact that $\mathrm{OMQA}(\mathcal{Q}, \mathcal{L})$ in combined complexity can be reduced to MINEX-REL$(\mathcal{Q}, \mathcal{L})$ in combined complexity for DLs $\mathcal{L}$ containing $\mathcal{EL}$. To see why this holds, it is sufficient to observe that $\mathcal{EL}$ contains conjunctions on the left-hand side. As a result, if we have an instance query $Q = C(a)$, an ABox $\mathcal{A}$, and a TBox $\mathcal{T}$, we take two new concept symbols D and F, and add the axiom $C \sqcap D \sqsubseteq F$ to $\mathcal{T}$, call it $\mathcal{T}'$. Then, $\mathcal{A} \models (Q, \mathcal{T})$ iff there is a MINEX for $(F(a), \mathcal{T}')$ in $\mathcal{A} \cup \{D(a)\}$ containing $D(a)$. If Q is a UCQ, a similar reduction applies: simply take a fresh concept symbol D, and add $D(a)$ to every conjunct of the given UCQ Q, and call it Q'. Note that $\mathcal{A} \models (Q, \mathcal{T})$ iff there is a MINEX for $(Q', \mathcal{T})$ in $\mathcal{A} \cup \{D(a)\}$ containing $D(a)$.

5 RELATED WORK

The task of deriving explanations for query answers is a form of logical abduction [36, 19]. Explanations are first considered in the context of DLs with the work of [32], which is then followed by [10]. The main goal of these early works in DLs was to explain *why* a concept subsumption is entailed from a given theory, which amounts to finding proofs for a given concept subsumption, based on the underlying deductive calculi.

Viewing minimal subsets of the TBox as explanations (instead of the specific proofs) represents a shift in the literature [38]. This approach is often referred to as *axiom pinpointing* [25, 4, 34], and such explanations are called *justifications* [23, 24]. This line of research has received much attention, and a number of systems have been developed [25, 39].

The problem of ABox abduction [28] is closer to our approach in the sense that it views sets of assertions as explanations, as opposed to sets of TBox axioms. However, the problem studied in [28], as well as in [14], is very different from ours: Given an instance query that is *not* entailed from the knowledge base, the idea is to find a set of assertions such that, when they are added to the ABox, the entailment holds. On the other hand, our explanations are for entailments that are known to hold, and any minimal *subset* of the ABox that satisfies the query is an explanation.

To our knowledge, explanations for (positive) OMQA are only studied for the *DL-Lite* family of languages [11] in the context of DLs. Our work builds on the recent literature on explanations investigated for existential rules [16]. This work lifts the ideas from the axiom pinpointing literature [25, 4, 34] to OMQA and introduces the decision problems studied in this paper. Our focus is on DLs, and our results differ significantly: All the results reported in this paper are novel, except for the membership results that are explicitly given as corollaries to earlier work [16]. In particular, our hardness results are in a stronger form, as they are given for binary arity. Besides, DLs that allow disjunction require new techniques as compared to existential rules (see, e.g., our results on $\mathcal{ALC}$).

Other related work is by [7], where the idea is to explain query answers under inconsistency-tolerant semantics. The notion of MINEX can be found as *minimal $\mathcal{T}$-support* [8] and a *cause* [7] in this context. These approaches are clearly incomparable to our approach, as they are relative to a different semantics, which also implies very different complexity results. Finally, we note an earlier work on deriving explanations for OMQs under existential rules [15], which considers probabilistic databases as the data model and hence is of a different flavor.

6 SUMMARY AND OUTLOOK

We studied the problem of explaining query answers in terms of minimal subsets of the ABox assertions, and provided a thorough complexity analysis for several decision problems associated with minimal explanations. There are a number of future research directions. It would be interesting to do a more fine-grained complexity analysis to better identify queries that are tractable, e.g., such classification results are obtained, e.g., for classical $\mathrm{OMQA}(\mathrm{IQ}, \mathcal{EL})$ [30], and for consistent query answering over a restricted class of queries [31].

We also want to study other variants of explanation-related problems, such as counting the number of minimal explanations. In most cases, it is easy to see that such problems are #P-hard [34]. In our framework, counting-related explanation problems are necessarily related to existing results on OMQA over probabilistic databases [12]. We also note the recent dichotomy result for query evaluation over probabilistic data over binary signatures [2]. We expect many of the existing techniques from probabilistic query answering to translate to the problem of counting the number of minimal explanations.

Similarly to axiom pinpointing [34], it is worthwhile studying enumeration-related problems, i.e., understanding the complexity of enumerating all MinAs w.r.t. a specific order. The results in this paper form the basis for the study of enumeration problems; a detailed treatment of counting and enumeration problems is future work.

ACKNOWLEDGMENTS

We thank the anonymous reviewers of this paper, who provided very useful feedback. This work was supported by the Alan Turing Institute under the UK EPSRC grant EP/N510129/1, by the AXA Reseach Fund, and by the EPSRC grants EP/R013667/1, EP/L012138/1, and EP/M025268/1.

REFERENCES

[1] Andrea Acciarri, Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Mattia Palmieri, and Riccardo Rosati, 'Quonto: Querying ontologies', in *Proc. AAAI*, pp. 1670–1671, (2005).
[2] Antoine Amarilli and İsmail İlkan Ceylan, 'A dichotomy for homomorphism-closed queries on probabilistic graphs', in *Proc. of ICDT*, (2020).
[3] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev, 'The DL-Lite family and relations', *J. Artif. Intell. Res.*, **36**, 1–69, (2009).
[4] Franz Baader and Boontawee Boontawee Suntisrivaraporn, 'Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}^+$', in *Proc. KR-MED*, (2008).
[5] *The Description Logic Handbook: Theory, Implementation, and Applications*, eds., Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, Cambridge University Press, 2nd edn., 2007.
[6] Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn, 'Pinpointing in the description logic $\mathcal{EL}^+$', in *Proc. KI*, pp. 52–67, (2007).
[7] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué, 'Computing and explaining query answers over inconsistent DL-Lite knowledge bases', *J. Artif. Intell. Res.*, **64**, 563–644, (2019).
[8] Meghyn Bienvenu and Riccardo Rosati, 'Tractable approximations of consistent query answering for robust ontology-based data access', in *Proc. IJCAI*, pp. 775–781, (2013).
[9] Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter, 'Ontology-based data access: A study through Disjunctive Datalog, CSP, and MMSNP', *ACM Trans. Database Syst.*, **39**(4), 33:1–33:44, (2014).
[10] Alex Borgida, Enrico Franconi, and Ian Horrocks, 'Explaining $\mathcal{ALC}$ subsumption', in *Proc. ECAI*, pp. 209–213, (2000).
[11] Alexander Borgida, Diego Calvanese, and Mariano Rodriguez-Muro, 'Explanation in the DL-Lite family of description logics', in *Proc. OTM*, pp. 1440–1457, (2008).
[12] Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz, 'Ontology-mediated queries for probabilistic databases', in *Proc. of AAAI*, pp. 1063–1069, (2017).
[13] Endre Boros, Vladimir Gurvich, Leonid Khachiyan, and Kazuhisa Makino, 'On maximal frequent and minimal infrequent sets in binary matrices', *Ann. Math. Artif. Intell.*, **39**(3), 211–221, (2003).
[14] Diego Calvanese, Magdalena Ortiz, Mantas Šimkus, and Giorgio Stefanoni, 'Reasoning about explanations for negative query answers in dl-lite', *J. Artif. Intell. Res.*, **48**(1), 635–669, (2013).
[15] İsmail İlkan Ceylan, Stefan Borgwardt, and Thomas Lukasiewicz, 'Most probable explanations for probabilistic database queries', in *Proc. IJCAI*, pp. 950–956, (2017).
[16] İsmail İlkan Ceylan, Thomas Lukasiewicz, Enrico Malizia, and Andrius Vaicenavičius, 'Explanations for query answers under existential rules', in *Proc. IJCAI*, pp. 1639–1646, (2019).
[17] Gene Ontology Consortium, 'The gene ontology (GO) database and informatics resource', *Nucleic Acids Res.*, **32**(Database-Issue), 258–261, (2004).
[18] Ronald Cornet and Nicolette de Keizer, 'Forty years of SNOMED: a literature review', *BMC Med. Inf. & Decision Making*, **8**(S-1), S2, (2008).
[19] Thomas Eiter and Georg Gottlob, 'The complexity of logic-based abduction', *J. ACM*, **42**(1), 3–42, (1995).
[20] H. Gabow, S. Maheshwari, and L. Osterweil, 'On two problems in the generation of program test paths', *IEEE Trans. Soft. Eng.*, 227–231, (1976).
[21] Andrew Gainer-Dewar and Paola Vera-Licona, 'The minimal hitting set generation problem: Algorithms and computation', *SIAM J. Discr. Math.*, **31**(1), 63–100, (2017).
[22] Michael R. Garey and David S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., 1990.
[23] Matthew Horridge, Bijan Parsia, and Ulrike Sattler, 'Laconic and precise justifications in OWL', in *Proc. ISWC*, pp. 323–338, (2008).
[24] Matthew Horridge, Bijan Parsia, and Ulrike Sattler, 'Explaining inconsistencies in OWL ontologies', in *Proc. SUM*, pp. 124–137, (2009).
[25] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin, 'Finding all justifications of OWL DL entailments', in *Proc. ISWC*, pp. 267–280, (2007).
[26] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and Bernardo Cuenca-Grau, 'Repairing unsatisfiable concepts in OWL ontologies', in *Proc. ESWC*, pp. 170–184, (2006).
[27] Steffen Klamt and Ernst Dieter Gilles, 'Minimal cut sets in biochemical reaction networks', *Bioinformatics*, **20**(2), 226–234, (2004).
[28] Szymon Klarman, Ulle Endriss, and Stefan Schlobach, 'ABox abduction in the description logic $\mathcal{ALC}$', *J. Auto. Reas.*, **46**(1), 43–80, (2011).
[29] Andrea S. Lapaugh and Christos H. Papadimitriou, 'The even-path problem for graphs and digraphs', *Networks*, **14**(4), 507–513, (1984).
[30] Carsten Lutz and Leif Sabellek, 'Ontology-mediated querying with $\mathcal{EL}$: Trichotomy and linear datalog rewritability', in *Proc. of IJCAI*, (2017).
[31] Carsten Lutz and Frank Wolter, 'On the relationship between consistent query answering and constraint satisfaction problems', in *Proc. of ICDT*, pp. 363–379, (2015).
[32] Deborah L. McGuinness and Alexander Borgida, 'Explaining subsumption in description logics', in *Proc. IJCAI*, pp. 816–821, (1995).
[33] Christos H. Papadimitriou and David Wolfe, 'The complexity of facets resolved', *J. Comput. Syst. Sci.*, **37**(1), 2 – 13, (1988).
[34] Rafael Peñaloza and Barış Sertkaya, 'Understanding the complexity of axiom pinpointing in lightweight description logics', *Artif. Intell.*, **250**, 80 – 104, (2017).
[35] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, 'Linking data to ontologies', in *J. Data Sem. X*, volume 4900 of *LNCS*, 133–173, Springer-Verlag, (2008).
[36] Raymond Reiter, 'A theory of diagnosis from first principles', *Artif. Intell.*, **32**(1), 57 – 95, (1987).
[37] Marcus Schaefer and Christopher Umans, 'Completeness in the polynomial-time hierarchy: A compendium', *SIGACT News*, (2002).
[38] Stefan Schlobach and Ronald Cornet, 'Non-standard reasoning services for the debugging of description logic terminologies', in *Proc. IJCAI*, pp. 355–360, (2003).
[39] Roberto Sebastiani and Michele Vescovi, 'Axiom pinpointing in lightweight description logics via Horn-SAT encoding and conflict analysis', in *Proc. CADE*, (2009).
[40] Larry J. Stockmeyer, 'The polynomial-time hierarchy', *Theor. Comput. Sci.*, **3**(1), 1 – 22, (1976).
[41] Boontawee Suntisrivaraporn, Guilin Qi, Qiu Ji, and Peter Haase, 'A modularization-based approach to finding all justifications for OWL DL entailments', in *Proc. ASWC*, pp. 1–15, (2008).
[42] Moshe Y. Vardi, 'The complexity of relational query languages (extended abstract)', in *Proc. STOC*, pp. 137–146, (1982).
[43] Celia Wrathall, 'Complete sets and the polynomial-time hierarchy', *Theor. Comput. Sci.*, **3**(1), 23 – 33, (1976).
[44] Yujiao Zhou, Bernardo Cuenca Grau, Yavor Nenov, Mark Kaminski, and Ian Horrocks, 'PAGOdA: Pay-as-you-go ontology query answering using a datalog reasoner', *J. Artif. Intell. Res.*, **54**, 309–367, (2015).