Explanations for Negative Query Answers under Existential Rules*

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

[1]Department of Computer Science, University of Oxford, UK
[2]Institute of Logic and Computation, TU Wien, Austria
[3]DISI, University of Bologna, Italy
[4]DIMES, University of Calabria, Italy

Abstract
Ontology-mediated query answering is an extensively studied paradigm, where the conceptual knowledge provided by an ontology is leveraged towards more enhanced querying of data sources. A major advantage of ontological reasoning is its interpretability, which allows one to derive explanations for query answers. Indeed, explanations have a long history in knowledge representation, and have also been investigated for ontology languages based on description logics and existential rules. Existing works on existential rules, however, merely focus on understanding why a query is entailed, i.e., explaining positive query answers. In this paper, we continue this line of research and address another important problem, namely, explaining why a query is *not* entailed under existential rules, i.e., explaining negative query answers. We consider various problems related to explaining non-entailments from the abduction literature, and also introduce new problems. For all considered problems, we give a detailed complexity analysis for a wide range of existential rule languages and complexity measures.

Keywords
Ontologies, Existential rules, Negative query answering, Explanations, Computational complexity

1. Introduction

Ontology-based query answering enhances querying of data sources with an ontology encoding domain knowledge. The idea is to view the ontology and the user query as a composite query, called *ontology-mediated query (OMQ)*, and the task of evaluating such queries is called *ontology-mediated query answering (OMQA)* [2]. OMQA is an important paradigm in knowledge representation with many application areas. Description logics (DLs) [3] and existential rules [4, 5] are two families of languages commonly used to formulate ontologies.

Discussion Papers - 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022)
*This is an abridged version of the KR 2020 paper [1].
ismail.ceylan@cs.ox.ac.uk (İ. İ. Ceylan); thomas.lukasiewicz@cs.ox.ac.uk (T. Lukasiewicz); enrico.malizia@unibo.it (E. Malizia); cmolinaro@dimes.unical.it (C. Molinaro); andrius.vaicenavicius@cs.ox.ac.uk (A. Vaicenavičius)
0000-0003-4118-4689 (İ. İ. Ceylan); 0000-0002-7644-1668 (T. Lukasiewicz); 0000-0002-6780-4711 (E. Malizia); 0000-0003-4103-1084 (C. Molinaro)

With the increasing demand for more explainable systems, explanations for OMQA have recently seen a surge in interest. The most basic problem is explaining why a query is entailed, i.e., explaining *positive* query answers. This problem has been studied for ontology languages based on DLs [6, 7, 8, 9] and existential rules [10]. The main idea is viewing an explanation as a set of database facts, which, together with the ontology, are sufficient to entail the query.

The complementary problem of explaining why a query is *not* entailed, i.e., explaining *negative* query answers, has only been studied for DLs [11], where the problem is modeled as an abduction task. Abduction has been studied for several formalisms, such as propositional logic [12], logic programs [13], default theories [14], probabilistic temporal logic [15], and DLs [16, 17, 11]. The closest work to ours is [11]: for a given query that is *not* entailed from the knowledge base, find a set of assertions (avoiding inconsistencies) such that, when added to the ABox, the entailment holds.

In our paper, we continue this line of research, and address the problem of explaining negative answers in OMQA based on existential rules (rather than DLs) as underlying ontology languages. We provide a precise complexity picture of various computational tasks for a wide range of existential rule languages and under different complexity measures.

2. Preliminaries

We briefly recall some basics on existential rules from the context of Datalog$^{\pm}$ [5].

General. We assume a set $\mathbf{C}$ of *constants*, a set $\mathbf{N}$ of *labeled nulls*, and a set $\mathbf{V}$ of *variables*. A *term* t is a constant, null, or variable. We assume a set of *predicates*, each associated with an arity. An *atom* has the form $p(t_1, \ldots, t_n)$, where p is an n-ary predicate, and $t_1, \ldots, t_n$ are terms. An atom containing only constants is called *fact*. Conjunctions of atoms are also identified with the sets of their atoms. An *instance* I is a (possibly infinite) set of atoms defined over constants and nulls. A *database* D is a finite instance containing only constants. A *homomorphism* is a substitution $h\colon D \cup \mathbf{N} \cup \mathbf{V} \mapsto D \cup \mathbf{N} \cup \mathbf{V}$ that is the identity on D and maps $\mathbf{N}$ to $D \cup \mathbf{N}$. With a slight abuse of notation, homomorphisms are applied also to (sets/conjunctions of) atoms. A *conjunctive query* (CQ) Q has the form $\exists \mathbf{Y} \phi(\mathbf{X}, \mathbf{Y})$, where $\phi(\mathbf{X}, \mathbf{Y})$ is a conjunction of atoms without nulls. The *answer* to Q over an instance I, denoted $Q(I)$, is the set of all $|\mathbf{X}|$-tuples $\mathbf{t}$ over D for which there is a homomorphism h such that $h(\phi(\mathbf{X}, \mathbf{Y})) \subseteq I$ and $h(\mathbf{X}) = \mathbf{t}$. A *Boolean CQ* (BCQ) Q is a CQ $\exists \mathbf{Y} \phi(\mathbf{Y})$, i.e., all variables are existentially quantified; Q is *true* over I, denoted $I \models q$, if $Q(I) \neq \emptyset$, i.e., there is a homomorphism h with $h(\phi(\mathbf{Y})) \subseteq I$.

Dependencies. A *tuple-generating dependency (TGD)* σ is an FO formula $\forall \mathbf{X} \forall \mathbf{Y}\, \varphi(\mathbf{X}, \mathbf{Y}) \rightarrow \exists \mathbf{Z}\, p(\mathbf{X}, \mathbf{Z})$, where $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are pairwise disjoint sets of variables, $\varphi(\mathbf{X}, \mathbf{Y})$ is a conjunction of atoms, and $p(\mathbf{X}, \mathbf{Z})$ is an atom, all without nulls. Classes of TGDs are also known as *existential rules*, or *Datalog$^{\pm}$ languages* in the literature. An instance I satisfies σ, written $I \models \sigma$, whenever there exists a homomorphism h such that $h(\varphi(\mathbf{X}, \mathbf{Y})) \subseteq I$, then there exists $h' \supseteq h|_{\mathbf{X}}$, where $h|_{\mathbf{X}}$ is the restriction of h on $\mathbf{X}$, such that $h'(p(\mathbf{X}, \mathbf{Z})) \in I$. A *negative constraint (NC)* ν is a first-order formula $\forall \mathbf{X}\, \varphi(\mathbf{X}) \rightarrow \bot$, where $\mathbf{X} \subseteq \mathbf{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms without nulls, and $\bot$ denotes the truth constant *false*. An instance I satisfies ν, written $I \models \nu$, if there is *no* homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$. A *program* (or *ontology*) is a finite set Σ of TGDs and NCs. An instance I satisfies Σ, written $I \models \Sigma$, if I satisfies each TGD and NC of Σ.

Table 1
Complexity of BCQ answering under existential rules [20].

$\mathcal{L}$	**Data**	fp-**comb.**	ba-**comb.**	**Comb.**
L, LF, AF	in AC^0	NP	NP	PSPACE
S, SF	in AC^0	NP	NP	EXP
A	in AC^0	NP	NEXP	NEXP
G	P	NP	EXP	2EXP
F, GF	P	NP	NP	EXP
WS, WA	P	NP	2EXP	2EXP

For brevity, we omit the universal quantifiers in front of TGDs and NCs, and use the comma (instead of $\wedge$) for conjoining atoms. For a TGD class $\mathbb{C}$, $\mathbb{C}_\perp$ denotes the formalism obtained by combining $\mathbb{C}$ with arbitrary NCs.

The Datalog$^\pm$ languages $\mathcal{L}$ we consider guaranteeing decidability are among the most frequently analyzed in the literature, namely, linear (L) [5], guarded (G) [4], sticky (S) [18], and acyclic TGDs (A), along with the "weak" (proper) generalizations weakly sticky (WS) [18] and weakly acyclic TGDs (WA) [19], as well as their "full" (i.e., existential-free) proper restrictions linear full (LF), guarded full (GF), sticky full (SF), and acyclic full TGDs (AF), respectively, and full TGDs (F) in general. We also recall the following further inclusions: L $\subset$ G and F $\subset$ WA $\subset$ WS. We refer to [20] for a more detailed overview.

Ontology-Mediated Query Answering. An *ontology-mediated query (OMQ)* is a pair (Q, Σ), where Q is a query, and Σ is an ontology. Let $\mathcal{L}$ be a Datalog$^\pm$ language. If $\Sigma \in \mathcal{L}$, we say that (Q, Σ) is an $\mathcal{L}$-OMQ. For a database D and an OMQ (Q, Σ), the set $mods(D, \Sigma)$ of *models* of (D, Σ) is the set of instances $\{I \mid I \supseteq D \wedge I \models \Sigma\}$; D *entails* (Q, Σ), denoted $D \models (Q, \Sigma)$, if $I \models Q$ for every $I \in mods(D, \Sigma)$. A different way to define the existential rules semantics is via the concept of the *Chase* (see, e.g., [18, 21]). We say that (D, Σ) is *consistent* if $mods(D, \Sigma) \neq \emptyset$, otherwise (D, Σ) is *inconsistent*. *Ontology-mediated query answering (OMQA)* is the task of deciding whether $D \models (Q, \Sigma)$ for a given database D and an OMQ (Q, Σ). When OMQA($\mathcal{L}$) is restricted to the case where (D, Σ) is consistent, we talk of consistent-OMQA($\mathcal{L}$).

Following Vardi [22], the *combined complexity* of BCQ answering considers the database, the set of dependencies, and the query as part of the input. The *bounded-arity-combined* (or *ba-combined*) complexity assumes that the arity of the underlying schema is bounded by an integer constant. The *fixed-program-combined* (or *fp-combined*) *complexity* considers the sets of TGDs as fixed; the *data complexity* also assumes the query fixed. Table 1 summarizes the complexity results for OMQA in the different TGD classes here considered; OMQA($\mathcal{L}$) denotes the OMQA problem when restricted over ontologies belonging to $\mathcal{L}$.

An OMQ (Q, Σ) is *FO-rewritable*, if there exists a query Q_Σ such that, for all databases D, we have that $D \models (Q, \Sigma)$ iff $D \models Q_\Sigma$. In this case, Q_Σ is an *FO-rewriting* of (Q, Σ). A class of programs $\mathcal{L}$ is *FO-rewritable*, if it admits an FO-rewriting for every query and program in $\mathcal{L}$. All languages from Table 1 with AC^0 data complexity are FO-rewritable.

3. Explanations for Negative Query Answers

In this section, we formally define (minimal) explanations for negative query answers in OMQA along with several computational problems for them.

Definition 1. Let D be a database, let (Q, Σ) be an OMQ, with $D \not\models (Q, \Sigma)$, and let H be a finite set of facts. An *explanation* for $D \not\models (Q, \Sigma)$ w.r.t. H is a subset E of H such that $(D \cup E, \Sigma)$ is consistent and $D \cup E \models (Q, \Sigma)$. A *minimal explanation* (or *MinEX*) for $D \not\models (Q, \Sigma)$ w.r.t. H is an explanation E for $D \not\models (Q, \Sigma)$ w.r.t. H that is inclusion-minimal, i.e., no set $E' \subsetneq E$ is an explanation for $D \not\models (Q, \Sigma)$ w.r.t. H.

We now introduce the problems. A constraint on the input of all the problems is that the database D and the $\mathcal{L}$-OMQ (Q, Σ) are such that $D \not\models (Q, \Sigma)$.

The first problem is deciding whether a set of facts is a minimal explanation.

Problem: Is-MinEX$^{\not\models}(\mathcal{L})$.
Input: A database D, an $\mathcal{L}$-OMQ (Q, Σ), a finite set of facts H, and $E \subseteq H$.
Question: Is E a MinEX for $D \not\models (Q, \Sigma)$ w.r.t. H?

Another problem is deciding whether there exists a minimal explanation.

Problem: MinEX-Exists$^{\not\models}(\mathcal{L})$.
Input: A database D, an $\mathcal{L}$-OMQ (Q, Σ), and a finite set of facts H.
Question: Is there a MinEX for $D \not\models (Q, \Sigma)$ w.r.t. H?

Two other problems are recognizing relevant and necessary facts. A fact ψ is *relevant* (resp., *necessary*) for $D \not\models (Q, \Sigma)$ w.r.t. H iff ψ appears in at least one (resp., in every) MinEX for $D \not\models (Q, \Sigma)$ w.r.t. H.

Problem: MinEX-Rel$^{\not\models}(\mathcal{L})$.
Input: A database D, an $\mathcal{L}$-OMQ (Q, Σ), a finite set of facts H, and a fact ψ.
Question: Is ψ relevant for $D \not\models (Q, \Sigma)$ w.r.t. H?

Problem: MinEX-Nec$^{\not\models}(\mathcal{L})$.
Input: A database D, an $\mathcal{L}$-OMQ (Q, Σ), a finite set of facts H, and a fact ψ.
Question: Is ψ necessary for $D \not\models (Q, \Sigma)$ w.r.t. H?

The problems introduced so far are those commonly studied in the context of abductive reasoning and negative answer explanations (see, e.g., [11]). We here introduce two novel problems. The first asks whether a set H' of facts contains exactly all the relevant facts. This set is particularly interesting, as H' can be seen as a minimal over-approximation of all MinEXs, i.e., for every MinEX E, it holds that $E \subseteq H'$, and H' is the smallest set enjoying this property.

Problem: MinEX-AllRel$^{\not\models}(\mathcal{L})$.
Input: A database D, an $\mathcal{L}$-OMQ (Q, Σ), a finite set of facts H, and $H' \subseteq H$.
Question: Does H' contain exactly all the relevant facts for $D \not\models (Q, \Sigma)$ w.r.t. H?

The second novel problem that we consider asks whether a set H' contains exactly all the necessary facts. Interestingly, H' can be seen as a maximal under-approximation of all MinEXs, i.e., for every MinEX E, it holds that $H' \subseteq E$, and H' is the biggest set enjoying this property.

Table 2
Complexity of Is-MinEX$^{\not\models}(\mathcal{L})$ and MinEX-Exists$^{\not\models}(\mathcal{L})$.

	Is-MinEX$^{\not\models}(\mathcal{L})$				MinEX-Exists$^{\not\models}(\mathcal{L})$			
$\mathcal{L}$	**Data**	fp-**comb.**	ba-**comb.**	**Comb.**	**Data**	fp-**comb.**	ba-**comb.**	**Comb.**
$\mathsf{L}_\bot$, $\mathsf{LF}_\bot$, $\mathsf{AF}_\bot$	in P	D^{P}	D^{P}	PSPACE	in P	NP	Σ_2^{P}	PSPACE
$\mathsf{S}_\bot$, $\mathsf{SF}_\bot$	in P	D^{P}	D^{P}	EXP	in P	NP	Σ_2^{P}	EXP
$\mathsf{A}_\bot$	in P	D^{P}	D^{EXP}	D^{EXP}	in P	NP	P^{NEXP}	P^{NEXP}
$\mathsf{G}_\bot$	P	D^{P}	EXP	2EXP	NP	NP	EXP	2EXP
$\mathsf{F}_\bot$, $\mathsf{GF}_\bot$	P	D^{P}	D^{P}	EXP	NP	NP	Σ_2^{P}	EXP
$\mathsf{WS}_\bot$, $\mathsf{WA}_\bot$	P	D^{P}	2EXP	2EXP	NP	NP	2EXP	2EXP

Problem: MinEX-AllNec$^{\not\models}(\mathcal{L})$.
Input: A database D, an $\mathcal{L}$-OMQ (Q, Σ), a finite set of facts H, and $H' \subseteq H$.
Question: Does H' contain exactly all the necessary facts for $D \not\models (Q, \Sigma)$ w.r.t. H?

4. Is-MinEX$^{\not\models}$and MinEX-Exists$^{\not\models}$

We start with Is-MinEX$^{\not\models}(\mathcal{L})$, i.e., deciding whether a given set of facts is a minimal explanation for a negative query answer. The following theorem proves all upper bounds in Table 2. The intuition behind the result is: deciding whether a set E of facts is a MinEX requires to carry out essentially three tasks: (1) deciding whether $(D \cup E, \Sigma)$ is consistent; (2) deciding whether $D \cup E \models (Q, \Sigma)$; and (3) deciding whether E is inclusion-minimal.

Theorem 2. *For any language $\mathcal{L}$ here considered, if OMQA($\mathcal{L}$) is in $\mathcal{C}$ in the combined (resp., ba-comb., fp-comb., data) complexity, then Is-MinEX$^{\not\models}(\mathcal{L})$ can be decided with a $\mathcal{C}$ check and a linear number of co-$\mathcal{C}$ checks in the combined (resp., ba-comb., fp-comb., data) complexity.*

All the hardness results for Is-MinEX$^{\not\models}(\mathcal{L})$ in Table 2 descend from the hardness of deciding minimal explanations of positive query answers [10].

We now focus on MinEX-Exists$^{\not\models}(\mathcal{L})$, i.e., deciding the existence of (minimal) explanations for negative query answers. The following theorem proves all the upper bounds in Table 2, but the NP and P ones, that need tighter statements. Intuitively, to decide whether there exists a minimal explanation for a negative query answer, it suffices to check whether there is *any* explanations for the negative query answers, i.e., there is no need to double check the minimality.

Theorem 3. *For any language $\mathcal{L}$ here considered, if OMQA($\mathcal{L}$) is in $\mathcal{C}$ in the combined (resp., ba-comb., fp-comb., data) complexity, then MinEX-Exists$^{\not\models}(\mathcal{L})$ is in $\text{NP}^{\mathcal{C}}$ in the combined (resp., ba-comb., fp-comb., data) complexity.*

In the fp-combined setting, for the Datalog$^\pm$ languages here considered, checking whether a set of facts is consistent is feasible in P, because the negative constraints are fixed. This allows to obtain the following tighter result.

Theorem 4. *MinEX-Exists$^{\not\models}(\mathcal{L})$ is in NP in the fp-combined complexity for all the languages $\mathcal{L}$ here considered.*

Table 3
Complexity of MINEX-REL$^{\not\models}(\mathcal{L})$ and MINEX-ALLREL$^{\not\models}(\mathcal{L})$.

	MINEX-REL$^{\not\models}(\mathcal{L})$				MINEX-ALLREL$^{\not\models}(\mathcal{L})$			
$\mathcal{L}$	**Data**	fp-**comb.**	ba-**comb.**	**Comb.**	**Data**	fp-**comb.**	ba-**comb.**	**Comb.**
$\mathsf{L}_\perp$, $\mathsf{LF}_\perp$, $\mathsf{AF}_\perp$	in P	Σ_2^{P}	Σ_2^{P}	PSPACE	in P	$\mathrm{D}_2^{\mathrm{P}}$	$\mathrm{D}_2^{\mathrm{P}}$	PSPACE
$\mathsf{S}_\perp$, $\mathsf{SF}_\perp$	in P	Σ_2^{P}	Σ_2^{P}	EXP	in P	$\mathrm{D}_2^{\mathrm{P}}$	$\mathrm{D}_2^{\mathrm{P}}$	EXP
$\mathsf{A}_\perp$	in P	Σ_2^{P}	$\mathrm{P}^{\mathrm{NEXP}}$	$\mathrm{P}^{\mathrm{NEXP}}$	in P	$\mathrm{D}_2^{\mathrm{P}}$	$\mathrm{P}^{\mathrm{NEXP}}$	$\mathrm{P}^{\mathrm{NEXP}}$
$\mathsf{G}_\perp$	NP	Σ_2^{P}	EXP	2EXP	D^{P}	$\mathrm{D}_2^{\mathrm{P}}$	EXP	2EXP
$\mathsf{F}_\perp$, $\mathsf{GF}_\perp$	NP	Σ_2^{P}	Σ_2^{P}	EXP	D^{P}	$\mathrm{D}_2^{\mathrm{P}}$	$\mathrm{D}_2^{\mathrm{P}}$	EXP
$\mathsf{WS}_\perp$, $\mathsf{WA}_\perp$	NP	Σ_2^{P}	2EXP	2EXP	D^{P}	$\mathrm{D}_2^{\mathrm{P}}$	2EXP	2EXP

For FO-rewritable languages, the MinEXs for *positive* query answers when the query and the program are fixed can be computed in polynomial time [10]. By using this property, we can obtain the P upper bounds in Table 2.

Theorem 5. *If $\mathcal{L}$ is FO-rewritable language, then MINEX-EXISTS$^{\not\models}(\mathcal{L})$ is in P in the data complexity.*

For the hardness results, the NP-hardness results in the data complexity are via a reduction from SAT; the hardness results in the fp-comb., ba-comb., and combined complexity, but the Σ_2^{P}-hardness and the $\mathrm{P}^{\mathrm{NEXP}}$-hardness, are obtained via a reduction from OMQA($\mathcal{L}$) to MINEX-EXISTS$^{\not\models}(\mathcal{L})$; the Σ_2^{P}-hardness and the $\mathrm{P}^{\mathrm{NEXP}}$-hardness results are shown, respectively, via reductions from QBF validity and from the problem ETP [23]: given a triple (m, TP_1, TP_2), where m is a number in unary, and TP_1 and TP_2 are two tiling problems for the exponential square $2^n \times 2^n$, decide whether, for all initial tiling conditions w of length m, TP_1 has no solution with w *or* TP_2 has a solution with w.

5. MINEX-REL$^{\not\models}$ and MINEX-ALLREL$^{\not\models}$

We start by looking at the problem MINEX-REL$^{\not\models}(\mathcal{L})$ of deciding whether a fact is relevant. The following theorem proves all the upper bounds in Table 3, but those in the data complexity for FO-languages. Intuitively, to decide whether ψ is relevant, it suffices to guess a set E of facts containing ψ (feasible in NP), and then, via an oracle call, check that E is a minimal explanation.

Theorem 6. *For any language $\mathcal{L}$ here considered, if Is-MINEX$^{\not\models}(\mathcal{L})$ is in $\mathcal{C}$ in the combined (resp., ba-comb., fp-comb., data) complexity, then MINEX-REL$^{\not\models}(\mathcal{L})$ is in $\mathrm{NP}^{\mathcal{C}}$ in the combined (resp., ba-comb., fp-comb., data) complexity.*

By a consideration similar to that for Theorem 5 we obtain the following.

Theorem 7. *If $\mathcal{L}$ is FO-rewritable language, then MINEX-REL$^{\not\models}(\mathcal{L})$ is in P in the data complexity.*

All the hardness results for MINEX-REL$^{\not\models}(\mathcal{L})$ in Table 3 descends from the hardness of deciding the fact relevance in MinEXs of positive query answers [10].

We now analyze the problem MINEX-ALLREL$^{\not\models}(\mathcal{L})$ of deciding whether a set contains all and only the relevant facts. The following theorem proves all the upper bounds in Table 3.

Table 4
Complexity of MinEX-Nec$^{\not\models}(\mathcal{L})$ and MinEX-AllNec$^{\not\models}(\mathcal{L})$.

$\mathcal{L}$	MinEX-Nec$^{\not\models}(\mathcal{L})$ Data	*fp*-comb.	*ba*-comb.	Comb.	MinEX-AllNec$^{\not\models}(\mathcal{L})$ Data	*fp*-comb.	*ba*-comb.	Comb.
$\mathsf{L}_\perp$, $\mathsf{LF}_\perp$, $\mathsf{AF}_\perp$	$\leq$ P	co-NP	Π_2^P	PSPACE	$\leq$ P	D^P	D_2^P	PSPACE
$\mathsf{S}_\perp$, $\mathsf{SF}_\perp$	$\leq$ P	co-NP	Π_2^P	EXP	$\leq$ P	D^P	D_2^P	EXP
$\mathsf{A}_\perp$	$\leq$ P	co-NP	P^{NEXP}	P^{NEXP}	$\leq$ P	D^P	P^{NEXP}	P^{NEXP}
$\mathsf{G}_\perp$	co-NP	co-NP	EXP	2EXP	D^P	D^P	EXP	2EXP
$\mathsf{F}_\perp$, $\mathsf{GF}_\perp$	co-NP	co-NP	Π_2^P	EXP	D^P	D^P	D_2^P	EXP
$\mathsf{WS}_\perp$, $\mathsf{WA}_\perp$	co-NP	co-NP	2EXP	2EXP	D^P	D^P	2EXP	2EXP

Intuitively, H' is the set of all and only the relevant facts iff all facts in H' are relevant and all facts outside H' are not relevant.

Theorem 8. *For any language $\mathcal{L}$ here considered, if MinEX-Rel$^{\not\models}(\mathcal{L})$ is in $\mathcal{C}$ in the combined (resp., ba-comb., fp-comb., data) complexity, then MinEX-AllRel$^{\not\models}(\mathcal{L})$ can be decided with a $\mathcal{C}$ check and a co-$\mathcal{C}$ in the combined (resp., ba-comb., fp-comb., data) complexity.*

As for the hardness, the D^P-hardness and the D_2^P-hardness results are shown via reductions, respectively, from the Sat-Unsat problem and its generalization to the second level, i.e., decide the validity of two QBF formulas $\Phi = \exists X \forall Y \neg\phi(X, Y)$ and $\Psi = \forall X \exists Y \psi(X, Y)$ (to simplify the reduction, X and Y are assumed to be the same in ϕ and ψ [24]). The remaining hardness results are obtained via a reduction from MinEX-Exists$^{\not\models}(\mathcal{L})$ to MinEX-AllRel$^{\not\models}(\mathcal{L})$.

6. MinEX-Nec$^{\not\models}$ and MinEX-AllNec$^{\not\models}$

We now focus on the problem MinEX-Nec$^{\not\models}(\mathcal{L})$ of deciding whether a fact is necessary. The following theorem proves all upper bounds in Table 4, but the co-NP and P ones. Intuitively, we can *dis*prove that a fact ψ is necessary by checking that there is a (non-necessarily minimal) explanation excluding ψ.

Theorem 9. *For any language $\mathcal{L}$ here considered, if OMQA$(\mathcal{L})$ is in $\mathcal{C}$ in the combined (resp., ba-comb., fp-comb., data) complexity, then MinEX-Nec$^{\not\models}(\mathcal{L})$ is in co-$(\mathrm{NP}^{\mathcal{C}})$ in the combined (resp., ba-comb., fp-comb., data) complexity.*

By a consideration similar to that for Theorem 4 we obtain the following.

Theorem 10. *For any language $\mathcal{L}$ here considered, MinEX-Nec$^{\not\models}(\mathcal{L})$ is in* co-NP *in the fp-combined complexity.*

By a consideration similar to that for Theorem 5 we obtain the following.

Theorem 11. *If $\mathcal{L}$ is FO-rewritable language, then MinEX-Nec$^{\not\models}(\mathcal{L})$ is in* P *in the data complexity.*

The hardness results of $\textsc{MinEX-Nec}^{\not\models}(\mathcal{L})$ can be proven via a reduction from the complement of $\textsc{MinEX-Exists}^{\not\models}(\mathcal{L})$ to $\textsc{MinEX-Nec}^{\not\models}(\mathcal{L})$.

We now study the problem $\textsc{MinEX-AllNec}^{\not\models}(\mathcal{L})$ of deciding whether a set contains all and only the necessary facts. The following theorem proves all upper bounds in Table 4. Intuitively, H' is the set of all and only the necessary facts iff all facts in H' are necessary and all facts outside H' are not necessary.

Theorem 12. *For any language $\mathcal{L}$ here considered, if $\textsc{MinEX-Nec}^{\not\models}(\mathcal{L})$ is in $\mathcal{C}$ in the combined (resp., ba-comb., fp-comb., data) complexity, then $\textsc{MinEX-AllNec}^{\not\models}(\mathcal{L})$ can be decided by a check in $\mathcal{C}$ and a check in co-$\mathcal{C}$ in the combined (resp., ba-comb., fp-comb., data) complexity.*

As for the hardness, the $\textsc{d}^{\textsc{p}}$-hardness and the $\textsc{d}_2^{\textsc{p}}$-hardness results are shown via reductions from the $\textsc{Sat-Unsat}$ problem and its second level generalization (see above), respectively. The remaining hardness results are proven via a reduction from $\textsc{MinEX-Exists}^{\not\models}(\mathcal{L})$ to the complement of $\textsc{MinEX-AllNec}^{\not\models}(\mathcal{L})$.

7. Summary and Outlook

We have addressed the problem of explaining why a query is not entailed in OMQA under existential rules. We have conducted a detailed complexity analysis for various explanation problems, for a wide range of existential rule languages and under different complexity measures.

Our work on explaining OMQA has been extended to the inconsistent setting [25, 26] and to preferred explanations [27]. The explanation notion, in line with the inconsistent setting above, can also be extended to the cases of cardinality-based repairs [28], generalized repairs [29], preferred repairs [30, 31], probabilistic approaches [32, 33], and repairs based on value updates [34]. Inspired by the idea of exploring preferences over explanations, we can also consider how more elaborate preference models can be included in this framework [35, 36, 37, 38]. Another interesting direction for future work is actually computing all explanations or relevant/necessary facts.

Acknowledgments

This work was supported by the Alan Turing Institute under the UK EPSRC grant EP/N510129/1, the AXA Research Fund, and the EPSRC grants EP/R013667/1, EP/L012138/1, and EP/M025268/1.

References

[1] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicenavičius, Explanations for negative query answers under existential rules, in: Proc. KR, 2020, pp. 223–232.

[2] M. Bienvenu, B. T. Cate, C. Lutz, F. Wolter, Ontology-based data access: A study through disjunctive Datalog, CSP, and MMSNP, ACM T. Database Syst. 39 (2014) 33:1–33:44.

[3] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook, 2nd ed., Cambridge University Press, 2007.

[4] A. Calì, G. Gottlob, M. Kifer, Taming the infinite chase: Query answering under expressive relational constraints, J. Artif. Intell. Res. 48 (2013) 115–174.

[5] A. Calì, G. Gottlob, T. Lukasiewicz, A general Datalog-based framework for tractable query answering over ontologies, J. Web Sem. 14 (2012) 57–83.
[6] M. Horridge, B. Parsia, U. Sattler, Laconic and precise justifications in OWL, in: Proc. ISWC, 2008, pp. 323–338.
[7] R. Peñaloza, B. Sertkaya, Understanding the complexity of axiom pinpointing in lightweight description logics, Artif. Intell. 250 (2017) 80–104.
[8] A. Borgida, D. Calvanese, M. Rodriguez-Muro, Explanation in the DL-Lite family of description logics, in: Proc. OTM, 2008, pp. 1440–1457.
[9] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Explanations for ontology-mediated query answering in description logics, in: Proc. ECAI, 2020, pp. 672–679.
[10] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Explanations for query answers under existential rules, in: Proc. IJCAI, 2019, pp. 1639–1646.
[11] D. Calvanese, M. Ortiz, M. Šimkus, G. Stefanoni, Reasoning about explanations for negative query answers in DL-Lite, J. Artif. Intell. Res. 48 (2013) 635–669.
[12] T. Eiter, G. Gottlob, The complexity of logic-based abduction, J. ACM 42 (1995) 3–42.
[13] T. Eiter, G. Gottlob, N. Leone, Semantics and complexity of abduction from default theories, Artif. Intell. 90 (1997) 177–223.
[14] T. Eiter, G. Gottlob, N. Leone, Abduction from logic programs: Semantics and complexity, Theor. Comput. Sci. 189 (1997) 129–177.
[15] C. Molinaro, A. Sliva, V. S. Subrahmanian, Super-solutions: Succinctly representing solutions in abductive annotated probabilistic temporal logic, ACM Trans. Comput. Log. 15 (2014) 18:1–18:35.
[16] J. Du, K. Wang, Y. Shen, A tractable approach to ABox abduction over description logic ontologies, in: Proc. AAAI, 2014, pp. 1034–1040.
[17] S. Klarman, U. Endriss, S. Schlobach, ABox abduction in the description logic $\mathcal{ALC}$, J. Autom. Reasoning 46 (2011) 43–80.
[18] A. Calì, G. Gottlob, A. Pieris, Towards more expressive ontology languages: The query answering problem, Artif. Intell. 193 (2012) 87–128.
[19] R. Fagin, P. G. Kolaitis, R. J. Miller, L. Popa, Data exchange: Semantics and query answering, Theor. Comput. Sci. 336 (2005) 89–124.
[20] T. Lukasiewicz, E. Malizia, M. V. Martinez, C. Molinaro, A. Pieris, G. I. Simari, Inconsistency-tolerant query answering for existential rules, Artif. Intell. 307 (2022) art. no. 103685.
[21] E. Tsamoura, D. Carral, E. Malizia, J. Urbani, Materializing knowledge bases via trigger graphs, Proc. VLDB Endow. 14 (2021) 943–956.
[22] M. Y. Vardi, The complexity of relational query languages, in: Proc. STOC, 1982, pp. 137–146.
[23] T. Eiter, T. Lukasiewicz, L. Predoiu, Generalized consistent query answering under existential rules, in: Proc. KR, 2016, pp. 359–368.
[24] T. Lukasiewicz, E. Malizia, A novel characterization of the complexity class Θ_k^P based on counting and comparison, Theor. Comput. Sci. 694 (2017) 21–33.
[25] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for inconsistency-tolerant query answering under existential rules, in: Proc. AAAI, 2020, pp. 2909–2916.
[26] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for negative query answers under inconsistency-tolerant semantics, in: Proc. IJCAI, 2022, pp. 2705–2711.

[27] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicenavičius, Preferred explanations for ontology-mediated queries under existential rules, in: Proc. AAAI, 2021, pp. 6262–6270.
[28] T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Complexity of inconsistency-tolerant query answering in Datalog+/− under cardinality-based repairs, in: Proc. AAAI, 2019, pp. 2962–2969.
[29] T. Lukasiewicz, E. Malizia, C. Molinaro, Complexity of approximate query answering under inconsistency in Datalog$^{\pm}$, in: Proc. IJCAI, 2018, pp. 1921–1927.
[30] S. Flesca, F. Furfaro, F. Parisi, Preferred database repairs under aggregate constraints, in: Proc. SUM, 2007, pp. 215–229.
[31] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Preference-based inconsistency-tolerant query answering under existential rules, Artif. Intell. 312 (2022) art. no. 103772.
[32] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Query answering over inconsistent knowledge bases: A probabilistic approach, Theor. Comput. Sci. 935 (2022) 144–173.
[33] J. Grant, F. Parisi, A. Parker, V. S. Subrahmanian, An agm-style belief revision mechanism for probabilistic spatio-temporal logics, Artif. Intell. 174 (2010) 72–104.
[34] G. Alfano, S. Greco, C. Molinaro, I. Trubitsyna, An approximation algorithm for querying inconsistent knowledge bases, Intell. Syst. Appl. (2022) art. no. 200146.
[35] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On preferences and priority rules in abstract argumentation, in: Proc. IJCAI, 2022, pp. 2517–2524.
[36] T. Lukasiewicz, E. Malizia, Complexity results for preference aggregation over (m)CP-nets: Pareto and majority voting, Artif. Intell. 272 (2019) 101–142.
[37] T. Lukasiewicz, E. Malizia, Complexity results for preference aggregation over (m)CP-nets: Max and rank voting, Artif. Intell. 303 (2022) art. no. 103636.
[38] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, Non-transferable utility coalitional games via mixed-integer linear constraints, J. Artif. Intell. Res. 38 (2010) 633–685.