Explanations for Negative Query Answers under Inconsistency-Tolerant Semantics

Thomas Lukasiewicz[1,2] , **Enrico Malizia**[3] and **Cristian Molinaro**[4]
[1]Institute of Logic and Computation, TU Wien, Austria
[2]Department of Computer Science, University of Oxford, UK
[3]DISI, University of Bologna, Italy
[4]DIMES, University of Calabria, Italy
thomas.lukasiewicz@cs.ox.ac.uk, enrico.malizia@unibo.it, cmolinaro@dimes.unical.it

Abstract

Inconsistency-tolerant semantics have been proposed to provide meaningful query answers even in the presence of inconsistent knowledge. Recently, explainability has also become a prominent problem in different areas of AI. While the complexity of inconsistency-tolerant semantics is rather well-understood, not much attention has been paid yet to the problem of explaining query answers when inconsistencies may exist. Recent work on existential rules in the inconsistent setting has focused only on understanding why a query is entailed. In this paper, we address another important problem, which is explaining why a query is *not* entailed under an inconsistency-tolerant semantics. In particular, we consider three popular semantics, namely, the ABox repair, the intersection of repairs, and the intersection of closed repairs. We provide a thorough complexity analysis for a wide range of existential rule languages and for several complexity measures.

1 Introduction

In real ontology-based applications involving large amounts of data, possibly from different sources, inconsistency might naturally arise. To provide meaningful answers to users' queries even in such circumstances, different inconsistency-tolerant semantics of query answering have been proposed, mainly in the context of existential rules and description logics (DLs).

The *ABox repair* (AR) semantics is one of the most popular inconsistency-tolerant semantics; it was developed for relational databases [Arenas *et al.*, 1999] and generalized for several DLs [Lembo *et al.*, 2010]. Two other popular semantics introduced later on are the *intersection of repairs* (IAR) [Lembo *et al.*, 2010] and the *intersection of closed repairs* (ICR) [Bienvenu, 2012] semantics. Besides being AR semantics' natural under-approximations, they are relevant in practice as they are amenable to preprocessing, since the intersection of the (closed) repairs can be computed offline, and then standard query answering can be employed online. In fact, the latter approach has been used to implement the IAR semantics [Lembo *et al.*, 2015], while for the ICR semantics, it has been observed in [Bienvenu and Bourgaux, 2016].

The above semantics' complexity is rather well-understood in the literature (see, e.g., [Lembo *et al.*, 2010; Bienvenu, 2012; Bienvenu and Rosati, 2013; Bienvenu *et al.*, 2014; Lembo *et al.*, 2015; Bienvenu and Bourgaux, 2016] for inconsistency-tolerant query answering in DLs, and, e.g., [Eiter *et al.*, 2016; Lukasiewicz *et al.*, 2018; 2019; 2022] for inconsistency-tolerant query answering under existential rules).

Explainability has recently started to play a prominent role in different areas of AI. Explaining ontological query answers allows users to understand not only what is or is not entailed by a knowledge base under a particular semantics, but also *why*. It has recently been investigated under both DLs and existential rules [Arioua *et al.*, 2015; Bienvenu *et al.*, 2015; 2016; 2019; Hecham *et al.*, 2017; Ceylan *et al.*, 2019; 2020a; 2020b; 2021; Lukasiewicz *et al.*, 2020]. Bienvenu *et al.* [2015; 2016; 2019] considered the description logic DL-Lite$_\mathcal{R}$ and defined explanations for positive and negative answers under the brave, AR, and IAR semantics, providing a data complexity analysis of different related problems. Lukasiewicz *et al.* [2020] considered explanations of positive query answers in the inconsistent setting under existential rules. Although DLs are popular ontology formalisms, rule-based ones are well-suited for data intensive applications, as they admit higher-arity relations, naturally occurring in standard relational databases.

Explaining query answers under inconsistency-tolerant semantics includes the problem of explaining query *non*-entailment. Despite the importance of this problem, however, its complexity has received less attention in the literature so far. The only existing work focusing on this problem is [Bienvenu *et al.*, 2019]. In this paper, we continue this line of research. Rather than showing which facts inside the database are in conflict with the entailment of the query, as in [Bienvenu *et al.*, 2019], our notions of explanation are based on showing how fact removals, needed to restore consistency, cause the query non-entailment. Besides the AR and IAR semantics, we consider the ICR semantics as well. We carry out a thorough complexity analysis for a wide spectrum of Datalog$^\pm$ languages under the data, fixed-program-combined, bounded-arity-combined, and combined complexity measures.

2 Preliminaries

In this section, we briefly recall some basics on existential rules from the context of Datalog$^\pm$ [Calì *et al.*, 2012a].

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 also assume a set of *predicates*, each associated with an arity, i.e., a non-negative integer. 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 also called a *fact*. Conjunctions of atoms are often identified with the sets of their atoms. An *instance* I is a (possibly infinite) set of atoms $p(\mathbf{t})$, where $\mathbf{t}$ is a tuple of constants and nulls. A *database* D is a finite instance that contains only constants. A *homomorphism* is a substitution $h\colon \mathbf{C} \cup \mathbf{N} \cup \mathbf{V} \to \mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$ that is the identity on $\mathbf{C}$ and maps $\mathbf{N}$ to $\mathbf{C} \cup \mathbf{N}$. 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 tuples $\mathbf{t}$ over $\mathbf{C}$ 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; for BCQs, the only possible answer is the empty tuple. A BCQ 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 a first-order formula $\forall \mathbf{X} \forall \mathbf{Y}\, (\varphi(\mathbf{X}, \mathbf{Y}) \to \exists \mathbf{Z}\, p(\mathbf{X}, \mathbf{Z}))$, 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; $\varphi(\mathbf{X}, \mathbf{Y})$ is the *body* of σ, denoted *body*(σ), while $p(\mathbf{X}, \mathbf{Z})$ is the *head* of σ, denoted *head*(σ). For clarity, we consider single-atom-head TGDs; however, our results extend to TGDs with a conjunction of atoms in the head. An instance I satisfies σ, written $I \models \sigma$, if the following holds: 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 $\forall \mathbf{X}\, (\varphi(\mathbf{X}) \to \bot)$, where $\mathbf{X} \subseteq \mathbf{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms without nulls, called the *body* of ν and denoted *body*(ν), and $\bot$ denotes the truth constant *false*. An instance I satisfies ν, written $I \models \nu$, if there is *no* homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$. Given a set Σ of TGDs and NCs, I satisfies Σ, written $I \models \Sigma$, if I satisfies each TGD and NC of Σ. For brevity, we omit the universal quantifiers in front of TGDs and NCs, and use the comma (instead of $\wedge$) for conjoining atoms. Given a class of TGDs $\mathbb{C}$, we denote by $\mathbb{C}_\bot$ the formalism obtained by combining $\mathbb{C}$ with arbitrary NCs. Finite sets of TGDs and NCs are also called *programs*, and TGDs are also called *existential rules*.

The Datalog$^\pm$ languages that we consider to guarantee decidability are among the most frequently analyzed in the literature, namely, linear (L) [Calì *et al.*, 2012a], guarded (G) [Calì *et al.*, 2013], sticky (S) [Calì *et al.*, 2012b], and acyclic TGDs (A), along with the "weak" (proper) generalizations weakly sticky (WS) [Calì *et al.*, 2012b] and weakly acyclic TGDs (WA) [Fagin *et al.*, 2005], 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 [Lukasiewicz *et al.*, 2022] for a more detailed overview.

Knowledge Bases. A *knowledge base* is a pair (D, Σ), where D is a database, and Σ is a program. For a program Σ, Σ_T and Σ_{NC} denote the subsets of Σ containing the TGDs and NCs of Σ, respectively. The set of *models* of $KB = (D, \Sigma)$, denoted $mods(KB)$, is the set of instances $\{I \mid I \supseteq D \wedge I \models \Sigma\}$. We say that KB is *consistent* if $mods(KB) \neq \emptyset$, otherwise KB is *inconsistent*. The *answer* to a CQ q relative to KB is the set of tuples $ans(q, KB) = \bigcap\{q(I) \mid I \in mods(KB)\}$. The answer to a BCQ q is *true*, denoted $KB \models q$, if $ans(q, KB) \neq \emptyset$. A different, however equivalent, way to define ontological query answering is via the concept of the *Chase* (see, e.g., [Calì *et al.*, 2013; Tsamoura *et al.*, 2021], and references therein). The decision version of *CQ answering* is: given a knowledge base KB, a CQ q, and a tuple of constants $\mathbf{t}$, decide whether $\mathbf{t} \in ans(q, KB)$. Since CQ answering can be reduced in LOGSPACE to BCQ answering, we focus on BCQs. We denote by BCQ($\mathcal{L}$) the problem of BCQ answering when restricted over programs belonging to $\mathcal{L}$.

Following Vardi (1982), 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 and NCs as fixed; the *data complexity* also assumes the query fixed.

Table 1 recalls complexity results of BCQ answering for the languages considered in this paper [Lukasiewicz *et al.*, 2022].

Computational Complexity. For space reasons, we do not include preliminaries on computational complexity, which we assume the reader to be familiar with; we just recall that $\mathrm{D}_2^{\mathrm{P}} = \Sigma_2^{\mathrm{P}} \wedge \Pi_2^{\mathrm{P}}$ is the class of problems that are a conjunction of a problem in Σ_2^{P} and one in Π_2^{P}.

Inconsistency-Tolerant Semantics. Three prominent inconsistency-tolerant semantics for ontology-based query answering under existential rules are the *ABox repair (AR)* semantics, its approximation by the *intersection of repairs (IAR)*, and the *intersection of closed repairs (ICR)* semantics [Lembo *et al.*, 2010; Bienvenu, 2012]; all three are based on the notion of *repair*, which is a maximal consistent subset of the database.

Let $KB = (D, \Sigma)$ be a knowledge base. A *repair* of KB is an inclusion-maximal subset R of D such that $mods(R, \Sigma) \neq \emptyset$; $Rep(KB)$ denotes the set of all repairs of KB. In general, there might be inconsistent KBs *not* admitting any repair. This is the case when Σ itself is "incoherent", that is, there is no database B such that $mods(B, \Sigma) \neq \emptyset$—observe that, by monotonicity, this is equivalent to $mods(\emptyset, \Sigma) = \emptyset$. In such a case, even deleting the entire database would not be enough to gain back consistency. In the rest of the paper, we assume that the program of an inconsistent KB is never "incoherent". The *closure* $Cl(KB)$ of KB, which we denote also as $Cl(D, \Sigma)$, is the set of all facts built from constants in D and Σ, entailed by D and the TGDs of Σ. Let q be a BCQ:

- KB entails q under the *ABox repair* (AR) *semantics*, denoted $KB \models_{AR} q$, if $(R, \Sigma) \models q$ for all $R \in Rep(KB)$.
- KB entails q under the *intersection of repairs* (IAR) *semantics*, denoted $KB \models_{IAR} q$, if $(D_I, \Sigma) \models q$, where

	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

Table 1: Complexity of BCQ answering. All non-"in" entries are completeness results. [Lukasiewicz *et al.*, 2022]

$D_I = \bigcap\{R \mid R \in \mathit{Rep}(KB)\}$.

- KB entails q under the *intersection of closed repairs* (ICR) *semantics*, denoted $KB \models_{ICR} q$, if $(D_C, \Sigma) \models q$, where $D_C = \bigcap\{\mathit{Cl}(R, \Sigma) \mid R \in \mathit{Rep}(KB)\}$.

We refer to [Lukasiewicz *et al.*, 2018; 2022] for more on the complexity of AR-/IAR-/ICR-query answering.

3 Explanations for Negative Query Answers

In the rest of this section, $KB = (D, \Sigma)$ is a KB and q a BCQ.

The next definition recalls and extends the one of (minimal) explanation for query entailment by a KB [Ceylan *et al.*, 2019].

Definition 3.1. An *explanation* for q w.r.t. KB is a subset E of D such that (E, Σ) is consistent and $(E, \Sigma_T) \models q$.

A *minimal explanation* E, or *MinEx*, for q w.r.t. KB is an explanation for q w.r.t. KB that is inclusion-minimal, i.e., there is no $E' \subsetneq E$ that is an explanation for q w.r.t. KB.

Unlike the definition of (minimal) explanations in [Ceylan *et al.*, 2019], we require consistency, as in our setting KBs can be inconsistent. The above concept of minimal explanations is equivalent to that of *causes* in [Bienvenu *et al.* 2016; 2019].

Definition 3.2. A *repair deletion* of KB is a subset C of D such that $(D \setminus C, \Sigma)$ is consistent and, for every proper subset C' of C, $(D \setminus C', \Sigma)$ is inconsistent.

Note that minimality is included in the definition of repair deletion, which hence is a repair's complement; when the KB is consistent, the empty set is the only repair deletion.

Example 3.3. Consider the database

$$D = \{\mathit{Prof}(p), \mathit{Reader}(p), \mathit{Chair}(p), \mathit{Dean}(p), \mathit{PVC}(p)\},$$

asserting that p is a professor, a reader, chair (of some department), dean, and pro vice-chancellor. Consider also the program Σ consisting of the following TGDs Σ_T:

$$\begin{aligned}\Sigma_T = \{&\mathit{Prof}(X) \to \mathit{CanBeElectedFor}(X, 1),\\ &\mathit{Prof}(X), \mathit{Chair}(X), \to \mathit{CanBeElectedFor}(X, 3),\\ &\mathit{Reader}(X), \mathit{PVC}(X) \to \mathit{CanBeElectedFor}(X, 3),\\ &\mathit{CanBeElectedFor}(X, Y) \to \mathit{CanBeElected}(X)\}\end{aligned}$$

and the following NCs Σ_{NC}:

$$\begin{aligned}\Sigma_{NC} = \{&\mathit{Prof}(X), \mathit{Reader}(X) \to \bot,\\ &\mathit{Dean}(X), \mathit{Chair}(X) \to \bot,\\ &\mathit{Dean}(X), \mathit{Reader}(X) \to \bot\}.\end{aligned}$$

The TGDs assert that a professor can be elected for 1 year; a professor who is also chair, or a reader who is also pro vice-chancellor, can be elected for 3 years; and finally someone who can be elected for Y years can be elected. The NCs assert that one cannot be a professor and a reader at the same time, or dean and chair, and a reader cannot be dean.

The knowledge base $KB = (D, \Sigma)$ is inconsistent and admits the three repairs R_1, R_2, R_3, below, whose corresponding repair deletions are the sets C_1, C_2, C_3, respectively.

$$\begin{aligned}R_1 &= \{\mathit{PVC}(p), \mathit{Chair}(p), \mathit{Prof}(p)\}\\ R_2 &= \{\mathit{PVC}(p), \mathit{Chair}(p), \mathit{Reader}(p)\}\\ R_3 &= \{\mathit{PVC}(p), \mathit{Dean}(p), \mathit{Prof}(p)\}\\ C_1 &= \{\mathit{Reader}(p), \mathit{Dean}(p)\}\\ C_2 &= \{\mathit{Prof}(p), \mathit{Dean}(p)\}\\ C_3 &= \{\mathit{Reader}(p), \mathit{Chair}(p)\}\,.\end{aligned}$$

We now define the explanations for query non-entailment by a knowledge base under the AR, IAR, and ICR semantics.

Our explanation definitions aim at showing what happens in the reparation process, in terms of fact deletions, causing the query non-entailment under the considered inconsistency-tolerant semantics; to this aim, these explanations are given only in terms of pieces of MinExes for the query (lost during the reparation). From this perspective, our definitions and the ones by Bienvenu *et al.* [2019] are conceptually complementary, as the latter propose explanations in terms of database facts left untouched by the reparation process and that are inconsistent with the MinExes for the query; these explanations might not include pieces of the query MinExes at all. To give an example, to explain AR *non*-entailment, both explanation notions witness the presence of a repair R not entailing the query: Bienvenu *et al.* [2019] explanations provide facts F left in the repair R that are inconsistent with every MinEx for the query, whereas we provide facts E outside the repair R (deleted during the reparation) that intersect every MinEx for the query—here, F is a subset of a repair obtained by a deletion that is a superset of E.

In this respect, our notions of explanation for non-entailment under inconsistency-tolerant semantics are akin to the one defined for non-entailment of a query by a *consistent* knowledge base, as they both focus on "missing" facts needed to entail the query: in the consistent case, the missing facts were never in the database; in the inconsistent case, the missing facts become missing due to the reparation process' deletions.

The two perspectives can be thought to emerge from two symmetrical scenarios. In a first scenario, the query is not entailed when instead it was expected to be. In this case, knowing the facts left in the repair that are inconsistent with the MinExes, and hence preventing the query entailment, can be useful to "restore" the expected behavior of the KB (e.g., removing some of those facts might yield query entailment).

Symmetrically, there is the scenario in which the query is not entailed *and this was expected/desired.* In this context, knowing the facts discarded by the reparation process (whose deletion yields the desired non-entailment), as provided by our explanations, can be helpful.

Examples illustrating our notions of explanation are provided after their the formal definitions, which are now given.

Intuitively, a query q is entailed by a knowledge base under the AR semantics if every repair has a MinEx for q. An explanation for $KB \not\models_{AR} q$ provides a set C of facts witnessing the non-entailment of q by some repair: a minimal way to restore consistency needs to delete C from D, which leaves no MinEx for q in the repair yielded by the removal of (a superset of) C.

Definition 3.4. An *explanation for $KB \not\models_{AR} q$* is a set $\mathcal{E} = \{C\}$, where C is a subset of a repair deletion of KB such that, for every MinEx E for q w.r.t. KB, $E \cap C \neq \emptyset$.

Example 3.5. Consider the knowledge base KB of Example 3.3 and the query $q_1 = \mathit{CanBeElectedFor}(p, 3)$, asking whether p can be elected for 3 years. It is easy to see that q_1 is *not* entailed by KB under the AR semantics. An explanation for $KB \not\models_{AR} q_1$ is $\mathcal{E}_1 = \{C\}$, where $C = \{\mathit{Reader}(p), \mathit{Chair}(p)\}$. This explanation says that deleting (at least) C from D is one way to restore consistency that leaves no MinEx for q_1 (i.e., there is a repair not entailing q_1).

Intuitively, a query q is entailed by a knowledge base under the IAR semantics if the repairs have a MinEx for q in common. An explanation for $KB \not\models_{IAR} q$ provides sets of facts $C_1, \ldots, C_n$ explaining why no such common MinEx exists. In particular, whatever MinEx E for q is considered among all those in the database, a minimal way of restoring consistency needs to delete from D some C_i intersecting E, which is then lost in the repair yielded by the removal of (a superset of) C_i.

Definition 3.6. An *explanation for $KB \not\models_{IAR} q$* is a set $\mathcal{E} = \{C_1, \ldots, C_n\}$, with $n \geq 1$, where the C_is are subsets of repair deletions of KB such that, for every MinEx E for q w.r.t. KB, there exists a C_i such that $E \cap C_i \neq \emptyset$.

Example 3.7. Consider the knowledge base KB of Example 3.3 and the query $q_2 = \exists X\ \mathit{CanBeElected}(X)$, asking whether there is someone who can be elected. Query q_2 is *not* entailed by KB under the IAR semantics. An explanation for $KB \not\models_{IAR} q_2$ is $\mathcal{E}_2 = \{C_1, C_2\}$, where $C_1 = \{\mathit{Prof}(p)\}$ and $C_2 = \{\mathit{Reader}(p)\}$. This explanation says that deleting (at least) C_1 or C_2 from D are two ways to restore consistency; however, each MinEx for q_2 is lost either when deleting C_1 or C_2 (and thus there is no MinEx for q_2 common to all repairs).

Intuitively, a query q is entailed by a knowledge base under the ICR semantics if the *closures* of the repairs have a MinEx for q in common. An explanation for $KB \not\models_{ICR} q$ provides sets of facts $C_1, \ldots, C_n$ explaining why no such common MinEx exists. In particular, whatever MinEx E for q is considered among all those in the *closure of the database*, there is a minimal way of restoring consistency that needs to delete some C_i from D, and by doing so every way of deriving E is lost in the repair yielded by the removal of (a superset of) C_i.

Given a finite set of facts E, q_E denotes the BCQ $\bigwedge_{f \in E} f$.

Definition 3.8. An *explanation for $KB \not\models_{ICR} q$* is a set $\mathcal{E} = \{C_1, \ldots, C_n\}$, with $n \geq 1$, where the C_is are subsets of repair deletions of KB such that, for every MinEx E for q w.r.t. $(\mathit{Cl}(KB), \Sigma)$, there exists a C_i such that, for every MinEx X for q_E w.r.t. KB, $X \cap C_i \neq \emptyset$.

Note here that $(\mathit{Cl}(KB), \Sigma)$ might be inconsistent, however every MinEx E for q w.r.t. $(\mathit{Cl}(KB), \Sigma)$, as well as every MinEx X for q_E w.r.t. KB, must be consistent (by definition of MinEx). Nonetheless, there might be MinExes E for q w.r.t. $(\mathit{Cl}(KB), \Sigma)$ that do not admit any (consistent) MinEx for q_E; such a MinEx E, however, cannot belong to the intersection of the repairs' closures, because if there is no (consistent) MinEx for q_E, then E cannot appear in the closure of any repair.

Example 3.9. Consider Example 3.3's knowledge base KB and the query $q_3 = \exists X, Y\ \mathit{CanBeElectedFor}(X, Y)$, asking if there is someone who can be elected for some years, is *not* entailed by KB under the ICR semantics. The MinExes for q_3 in the closure of the database are $E_1 = \{\mathit{Prof}(p)\}$, $E_2 = \{\mathit{Reader}(p), \mathit{PVC}(p)\}$, $E_3 = \{\mathit{CanBeElectedFor}(p, 1)\}$, and $E_4 = \{\mathit{CanBeElectedFor}(p, 3)\}$. An explanation for $KB \not\models_{ICR} q_3$ is $\mathcal{E}_3 = \{C_1, C_2\}$, where $C_1 = \{\mathit{Prof}(p)\}$ and $C_2 = \{\mathit{Reader}(p), \mathit{Chair}(p)\}$. This explanation says that deleting (at least) C_1 or C_2 from D are two ways of restoring consistency; however, each E_i above cannot be derived anymore when deleting C_1 or C_2 (thus, there is no MinEx for q_3 common to all repair closures). In fact, E_1 and E_3 cannot be derived when C_1 is deleted, while E_2 and E_4 cannot be derived when C_2 is deleted. It is particularly interesting to note why E_4 is not derived when C_2 is deleted from D: the two (minimal) ways of deriving E_4, namely $X_1 = \{\mathit{Prof}(p), \mathit{Chair}(p)\}$ and $X_2 = \{\mathit{Reader}(p), \mathit{PVC}(p)\}$, are lost, and while the former is a MinEx for E_4, and hence an explanation for q_3, it is only a non-minimal explanation for q_3.

Definition 3.10. For each $S \in \{AR, IAR, ICR\}$, an explanation $\mathcal{E} = \{C_1, \ldots, C_n\}$ for $KB \not\models_S q$ is *minimal* iff there is no explanation for $KB \not\models_S q$ that can be derived from $\mathcal{E}$ by deleting facts from some C_i or by replacing a pair of distinct C_i and C_j with $C_i \cup C_j$.

The two conditions in the previous definition aims at making "minimal" both the facts in each C_i and the overall number of C_is in $\mathcal{E}$. Notice that if an explanation $\mathcal{E}$ contains a "redundant" C_i, that is, $\mathcal{E} \setminus \{C_i\}$ is an explanation, then $\mathcal{E}$ is not minimal according to the definition above, as we can delete all facts from C_i and still get an explanation, that is, $\mathcal{E}' = \mathcal{E} \setminus \{C_i\} \cup \{\emptyset\}$ is an explanation. In fact, even $\mathcal{E}'$ is not minimal, as the empty set can be merged with any other C_j in $\mathcal{E}'$, yielding the (possibly minimal) explanation $\mathcal{E} \setminus \{C_i\}$.

The next proposition states that explanations for query non-entailment by a KB under an inconsistency-tolerant semantics exist iff the query is not entailed under that semantics.

Proposition 3.11. *For each $S \in \{AR, IAR, ICR\}$, an explanation for $KB \not\models_S q$ exists iff $KB \not\models_S q$.*

A natural decision counterpart of the problem of computing minimal explanations for $KB \not\models_S q$, where $S \in \{AR, IAR, ICR\}$, is deciding the existence of a minimal explanation for $KB \not\models_S q$. In light of Proposition 3.11, the complexity of the latter problem equals the complexity of inconsistency-tolerant reasoning, which has already been investigated in the literature. Thus, in this paper, we focus on the following decision problems: deciding whether a set of sets of facts is a minimal explanation for $KB \not\models_S q$, for each $S \in \{AR, IAR, ICR\}$.

Problem: S-MinEx$^{\not\models}(\mathcal{L})$, with $S \in \{AR, IAR, ICR\}$.
Input: A knowledge base $KB = (D, \Sigma)$ with $\Sigma \in \mathcal{L}$, a BCQ q, and $\mathcal{E} \subseteq \mathcal{P}(D)$, with $\mathcal{P}(D)$ being the powerset of D.
Question: Is $\mathcal{E}$ a minimal explanation for $KB \not\models_S q$?

	Data	*fp*-**comb.**	*ba*-**comb.**	**Comb.**
$\mathsf{L}_\perp, \mathsf{LF}_\perp, \mathsf{AF}_\perp$	in P	D^P	D^P_2	PSPACE
$\mathsf{S}_\perp, \mathsf{SF}_\perp$	in P	D^P	D^P_2	EXP
$\mathsf{A}_\perp$	in P	D^P	P^NEXP	P^NEXP
$\mathsf{G}_\perp$	D^P	D^P	EXP	2EXP
$\mathsf{F}_\perp, \mathsf{GF}_\perp$	D^P	D^P	D^P_2	EXP
$\mathsf{WS}_\perp, \mathsf{WA}_\perp$	D^P	D^P	2EXP	2EXP

Table 2: Complexity of AR-/IAR-/ICR-MINEX$^{\not\models}$. All entries without "in" are completeness results. All these results hold also for deciding whether a set is a (not necessarily minimal) explanation.

In the following, for a knowledge base $KB = (D, \Sigma)$ and a query q, we assume that *(a)* $KB \not\models_S q$ and *(b)* D contains a MinEx for q, which is equivalent to require that there is at least one repair entailing the query—in the literature, this is the so-called *brave semantics*. Assumption *(a)* ensures that, if a set $\mathcal{E}$ is recognized as not being a minimal explanation for $KB \not\models_S q$, then this cannot be due to $KB \models_S q$. Assumption *(b)* rules out that $KB \not\models_S q$ holds simply because there is no way for the repairs (resp., their intersection, the intersection of their closure) to entail the query. It can be shown that the knowledge base does not contain any MinEx for q if and only if our *minimal* explanations are $\mathcal{E} = \{\emptyset\}$, for all semantics AR, IAR, and ICR. This is in line with our explanation definitions' spirit, as we want to explain how deletions cause the query non-entailment under the considered inconsistency-tolerant semantics; if the query is not entailed due to the absence of MinExes for it, then the inconsistency-tolerant semantics are not directly involved in the query non-entailment. *However, all the results in the paper hold even if we remove either of the aforementioned assumptions or both.*

4 Complexity Analysis

Our complexity results are summarized in Table 2. We first discuss membership, and then hardness results.

In the theorem statements, $\mathcal{L}$ *is any of the languages considered in this paper*, unless otherwise specified.

4.1 Membership Results

The following theorem provides upper bounds for the AR and IAR semantics. For each semantics, we employ a general procedure using an oracle for BCQ answering that applies to all languages and complexity measures considered here. The resulting upper bounds are always tight except for the *fp*-combined complexity of all languages and the data complexity of FO-rewritable languages (namely, $\mathsf{L}_\perp$, $\mathsf{A}_\perp$, $\mathsf{S}_\perp$, and their specializations), for which we need more refined statements.

Theorem 4.1. *If BCQ($\mathcal{L}$) is in the complexity class* $\mathbf{C}$ *in the combined (resp., ba-combined, fp-combined, data) complexity, then* AR-/IAR-MINEX$^{\not\models}$*($\mathcal{L}$) can be answered in the combined (resp., ba-combined, fp-combined, data) complexity by the following set of checks:*
(a) an $\text{NP}^{\mathbf{C}}$ *check, and*
(b) a co-($\text{NP}^{\mathbf{C}}$) *check.*

In the previous theorem, to decide whether a set $\mathcal{E} = \{C_1, \ldots, C_n\}$ is a minimal explanation for $KB \not\models_{IAR} q$, an $\text{NP}^{\mathbf{C}}$ check is needed to verify that each C_i is a subset of a repair deletion and no fact can be removed from any C_i (this ensures the minimality of each C_i by exploiting the concept of critical vertex in minimal hitting sets [Gottlob and Malizia, 2018], as a repair deletion is a minimal hitting set of all the culprits, which are minimal inconsistent subsets of the database), while a co-($\text{NP}^{\mathbf{C}}$) check is needed to verify that every MinEx for q intersects some C_i and no two distinct C_i and C_j can be merged into a single set. The procedure to answer AR-MINEX$^{\not\models}(\mathcal{L})$ is a special case where $\mathcal{E}$ is first checked to be a singleton, and then simpler conditions need to be verified (because of the presence of only one element in $\mathcal{E}$), which nonetheless yield the same upper bounds of IAR-MINEX$^{\not\models}(\mathcal{L})$. Membership in P^NEXP for $\mathsf{A}_\perp$ in the combined and *ba*-combined complexity (for both AR and IAR) follows from the collapse of the strong exponential hierarchy, i.e., for any $k \geq 0$, $(\Sigma^\text{P}_k)^\text{NEXP} = \text{P}^\text{NEXP}$ [Hemachandra, 1989]—this will also apply to the ICR semantics in Theorems 4.2 and 4.3.

Focusing on the ICR semantics, we employ, again, a general procedure that applies to all languages considered here. In contrast to the AR and IAR semantics, this procedure uses an oracle for BCQ answering *under the ICR semantics* and applies only to the combined complexity. For the other complexity measures, we devise different procedures. We denote by $ICR(\mathcal{L})$ the problem of BCQ answering under the ICR semantics when restricted over programs belonging to $\mathcal{L}$. The combined complexity of $ICR(\mathcal{L})$ is the same as the one reported in the last column of Table 2 [Lukasiewicz *et al.*, 2022].

Theorem 4.2. *If* $ICR(\mathcal{L})$ *is in the complexity class* $\mathbf{D}$ *in the combined complexity, then* ICR-MINEX$^{\not\models}$*($\mathcal{L}$) can be answered in the combined complexity by the following set of checks:*
(a) an $\text{NP}^{\mathbf{D}}$ *check, and*
(b) a co-($\text{NP}^{\mathbf{D}}$) *check.*

In the previous theorem, to decide whether a set $\mathcal{E} = \{C_1, \ldots, C_n\}$ is a minimal explanation for $KB \not\models_{ICR} q$, we can verify that each C_i is a subset of a repair deletion and no two distinct C_i and C_j can be merged into a single set in the same way as done for AR-/IAR-MINEX$^{\not\models}(\mathcal{L})$. The crucial difference is when we need to verify that *(a)* no fact can be removed from any C_i, which requires an $\text{NP}^{\mathbf{D}}$ check, and *(b)* for every MinEx E for q w.r.t. $(Cl(KB), \Sigma)$, there exists a C_i such that, for every MinEx X for q_E w.r.t. KB, $X \cap C_i \neq \emptyset$, which requires a co-($\text{NP}^{\mathbf{D}}$) check. A (naïve) procedure to verify the first condition (resp., the complement of the second one), would require guessing a MinEx for q w.r.t. $(Cl(KB), \Sigma)$, which can have exponential size, and such an approach would not yield tight upper bounds. To obtain upper bounds matching the lower ones, we reduce verifying such conditions to ICR reasoning over a suitable knowledge base.

In contrast, in the *ba*-combined (resp., *fp*-combined, data) complexity, a MinEx for q w.r.t. $(Cl(KB), \Sigma)$ can be guessed in NP, as $Cl(KB)$ is of polynomial size, since predicates' arities are bounded (resp., the program is fixed), which yields the following upper bounds for all languages considered.

Theorem 4.3. *If BCQ($\mathcal{L}$) is in the complexity class* $\mathbf{C}$ *in the ba-combined (resp., fp-combined, data) complexity, then*

ICR-MINEX$^{\not\models}$*($\mathcal{L}$) can be answered in the ba-combined (resp., fp-combined, data) complexity by the following set of checks:*
(a) an NP$^{\mathbf{C}}$ *check, and*
(b) a co-(NP$^{\mathbf{C}}$) *check.*

The results presented so far provide tight upper bounds for all languages we consider in the combined and *ba*-combined complexity. In the *fp*-combined complexity, for all languages and inconsistency-tolerant semantics, we need the following theorem to obtain tight upper bounds.

Theorem 4.4. *For $S \in \{AR, IAR, ICR\}$, S-MINEX$^{\not\models}$($\mathcal{L}$) is in* D$^{\text{P}}$ *in the fp-combined complexity.*

The membership results in the previous theorem derive from the same procedures used in the combined and *ba*-combined complexity with additional observations. First, consistency checking is in P in the *fp*-combined complexity. Second, to check query entailment, the certificate witnessing query entailment can be guessed by the Turing machine solving the original problem. Finally, for the *ICR*-MINEX$^{\not\models}$($\mathcal{L}$) problem, a MinEx for q w.r.t. $(Cl(KB), \Sigma)$ can be guessed in NP because $Cl(KB)$ has polynomial size.

The previous theorem provides tight upper bounds also in the data complexity for all languages but $\mathsf{L}_\perp$, $\mathsf{A}_\perp$, $\mathsf{S}_\perp$, and their specializations. For such languages, leveraging their FO-rewritability, we obtain the following tighter upper bound for all inconsistency-tolerant semantics.

Theorem 4.5. *For $S \in \{AR, IAR, ICR\}$, for $\mathcal{L} \in \{\mathsf{L}_\perp, \mathsf{A}_\perp, \mathsf{S}_\perp\}$, S-MINEX$^{\not\models}$($\mathcal{L}$) is in* P *in the data complexity.*

4.2 Hardness Results

Hardness results are stated for the most specific languages they apply to (and then hold for all generalizations, of course).

The following theorem establishes all D$^{\text{P}}$-hardness results in Table 2 in the data and *fp*-combined complexity for $\mathsf{GF}_\perp$ and its generalizations. We use a reduction from the classical D$^{\text{P}}$-complete problem SAT-UNSAT, that is, given two Boolean formulas ϕ and ψ in 3CNF, decide whether ϕ is satisfiable *and* ψ is unsatisfiable. For all inconsistency-tolerant semantics, from an instance of SAT-UNSAT, we derive a suitable knowledge base, a query, and a candidate minimal explanation $\mathcal{E} = \{C\}$. Roughly speaking, deciding satisfiability of ϕ is encoded into deciding whether C is a subset of a repair deletion, while deciding unsatisfiability of ψ is encoded into deciding whether C intersects all MinExes for the query.

Theorem 4.6. *For $S \in \{AR, IAR, ICR\}$, S-MINEX$^{\not\models}$($\mathsf{GF}_\perp$) is* D$^{\text{P}}$-hard *in the data complexity.*

The remaining D$^{\text{P}}$-hardness results in the *fp*-combined complexity are established by the following theorem, which holds by another (more involved) reduction from SAT-UNSAT.

Theorem 4.7. *For $S \in \{AR, IAR, ICR\}$, for $\mathcal{L} \in \{\mathsf{LF}_\perp, \mathsf{AF}_\perp, \mathsf{SF}_\perp\}$, S-MINEX$^{\not\models}$($\mathcal{L}$) is* D$^{\text{P}}$-hard *in the fp-combined complexity.*

The D$_2^{\text{P}}$-hardness results in Table 2 in the *ba*-combined complexity are established by the following theorem via a reduction from the prototypical D$_2^{\text{P}}$-hard problem: given two independent quantified Boolean formulas $\Phi = \exists X \forall Y \neg\phi(X, Y)$ and $\Psi = \forall X \exists Y \psi(X, Y)$, decide whether Φ and Ψ are both valid—here, ϕ and ψ are 3CNF formulas, and, to simplify the reduction even more, ϕ and ψ can also be assumed to have the same variables X and Y and the same number of clauses [Lukasiewicz and Malizia, 2017, Theorem 3.9].

Theorem 4.8. *For $S \in \{AR, IAR, ICR\}$, for $\mathcal{L} \in \{\mathsf{LF}_\perp, \mathsf{AF}_\perp, \mathsf{SF}_\perp\}$, S-MINEX$^{\not\models}$($\mathcal{L}$) is* D$_2^{\text{P}}$-hard *in the ba-combined complexity.*

The hardness results for $\mathsf{A}_\perp$ in the *ba*-combined and combined complexity require a dedicated analysis. We give a reduction from the following P$^{\text{NEXP}}$-complete problem [Lukasiewicz *et al.*, 2022]: given a triple (m, TP_1, TP_2), where m is a number in unary, and TP_1 and TP_2 are two instances of the tiling problem for the exponential square $2^n \times 2^n$, decide whether for every initial tiling condition w of length m, TP_1 has no solution with w *or* TP_2 has solution with w.

Theorem 4.9. *For $S \in \{AR, IAR, ICR\}$, S-MINEX$^{\not\models}$($\mathsf{A}_\perp$) is* P$^{\text{NEXP}}$-hard *in the ba-combined complexity.*

Finally, the remaining lower bounds in Table 2 in the *ba*-combined and combined complexity are established by reducing standard BCQ answering to our problems.

Theorem 4.10. *For $S \in \{AR, IAR, ICR\}$, for any language $\mathcal{L}$ without NCs considered in this paper, BCQ($\mathcal{L}$) is reducible in polynomial time to S-MINEX$^{\not\models}$($\mathcal{L}_\perp$) in the ba-combined, and combined complexity.*

5 Summary and Outlook

Explaining AI decisions and dealing with inconsistent knowledge have both garnered a lot of attention. In this paper, we have analyzed the problem of explaining query non-entailment by a knowledge base, providing a thorough complexity analysis for three popular inconsistency-tolerant semantics, a wide range of existential rules, and different complexity measures.

A direction for future work is analyzing the complexity of other related problems, such as deciding whether a fact is *necessary* (i.e., belonging to every minimal explanation) or *relevant* (i.e., belonging to at least one minimal explanation), as done by Bienvenu *et al.* [2016; 2019] for DL-Lite$_{\mathcal{R}}$. Also, it would be interesting to investigate other notions of explanations for query (non-)entailment, e.g., explanations providing facts together with NCs, or more general explanation notions encompassing the consistent and the inconsistent settings.

Acknowledgments

This work was supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1, by the AXA Research Fund, and by the EPSRC grant EP/R013667/1. We thank the anonymous reviewers for their insightful and helpful comments.

References

[Arenas *et al.*, 1999] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Consistent query answers in inconsistent databases. In *Proc. PODS*, pages 68–79, 1999.

[Arioua *et al.*, 2015] Abdallah Arioua, Nouredine Tamani, and Madalina Croitoru. Query answering explanation in inconsistent Datalog+/– knowledge bases. In *Proc. DEXA*, pages 203–219, 2015.

[Bienvenu and Bourgaux, 2016] Meghyn Bienvenu and Camille Bourgaux. Inconsistency-tolerant querying of description logic knowledge bases. In *Reasoning Web*, pages 156–202, 2016.

[Bienvenu and Rosati, 2013] Meghyn Bienvenu and Riccardo Rosati. Tractable approximations of consistent query answering for robust ontology-based data access. In *Proc. IJCAI*, pages 775–781, 2013.

[Bienvenu *et al.*, 2014] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Querying inconsistent description logic knowledge bases under preferred repair semantics. In *Proc. AAAI*, pages 996–1002, 2014.

[Bienvenu *et al.*, 2015] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Explaining query answers under inconsistency-tolerant semantics over description logic knowledge bases. In *Proc. DL*, 2015.

[Bienvenu *et al.*, 2016] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Explaining inconsistency-tolerant query answering over description logic knowledge bases. In *Proc. AAAI*, pages 900–906, 2016.

[Bienvenu *et al.*, 2019] 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.

[Bienvenu, 2012] Meghyn Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc. AAAI*, pages 705–711, 2012.

[Calì *et al.*, 2012a] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83, 2012.

[Calì *et al.*, 2012b] Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *Artif. Intell.*, 193:87–128, 2012.

[Calì *et al.*, 2013] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.*, 48:115–174, 2013.

[Ceylan *et al.*, 2019] İsmail İlkan Ceylan, Thomas Lukasiewicz, Enrico Malizia, and Andrius Vaicenavičius. Explanations for query answers under existential rules. In *Proc. IJCAI*, pages 1639–1646, 2019.

[Ceylan *et al.*, 2020a] İsmail İlkan Ceylan, Thomas Lukasiewicz, Enrico Malizia, Cristian Molinaro, and Andrius Vaicenavicius. Explanations for negative query answers under existential rules. In *Proc. KR*, pages 223–232, 2020.

[Ceylan *et al.*, 2020b] İsmail İlkan Ceylan, Thomas Lukasiewicz, Enrico Malizia, and Andrius Vaicenavicius. Explanations for ontology-mediated query answering in description logics. In *Proc. ECAI*, pages 672–679, 2020.

[Ceylan *et al.*, 2021] İsmail İlkan Ceylan, Thomas Lukasiewicz, Enrico Malizia, Cristian Molinaro, and Andrius Vaicenavicius. Preferred explanations for ontology-mediated queries under existential rules. In *Proc. AAAI*, pages 6262–6270, 2021.

[Eiter *et al.*, 2016] Thomas Eiter, Thomas Lukasiewicz, and Livia Predoiu. Generalized consistent query answering under existential rules. In *Proc. KR*, pages 359–368, 2016.

[Fagin *et al.*, 2005] Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

[Gottlob and Malizia, 2018] Georg Gottlob and Enrico Malizia. Achieving new upper bounds for the hypergraph duality problem through logic. *SIAM J. Comput.*, 47(2):456–492, 2018.

[Hecham *et al.*, 2017] Abdelraouf Hecham, Abdallah Arioua, Gem Stapleton, and Madalina Croitoru. An empirical evaluation of argumentation in explaining inconsistency-tolerant query answering. In *Proc. DL*, 2017.

[Hemachandra, 1989] Lane A. Hemachandra. The strong exponential hierarchy collapses. *J. Comput. Syst. Sci.*, 39(3):299–322, 1989.

[Lembo *et al.*, 2010] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Proc. RR*, pages 103–117, 2010.

[Lembo *et al.*, 2015] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant query answering in ontology-based data access. *J. Web Sem.*, 33:3–29, 2015.

[Lukasiewicz and Malizia, 2017] Thomas Lukasiewicz and Enrico Malizia. A novel characterization of the complexity class Θ_k^P based on counting and comparison. *Theor. Comput. Sci.*, 694:21–33, 2017.

[Lukasiewicz *et al.*, 2018] Thomas Lukasiewicz, Enrico Malizia, and Cristian Molinaro. Complexity of approximate query answering under inconsistency in Datalog+/–. In *Proc. IJCAI*, pages 1921–1927, 2018.

[Lukasiewicz *et al.*, 2019] Thomas Lukasiewicz, Enrico Malizia, and Andrius Vaicenavičius. Complexity of inconsistency-tolerant query answering in Datalog+/– under cardinality-based repairs. In *Proc. AAAI*, pages 2962–2969, 2019.

[Lukasiewicz *et al.*, 2020] Thomas Lukasiewicz, Enrico Malizia, and Cristian Molinaro. Explanations for inconsistency-tolerant query answering under existential rules. In *Proc. AAAI*, pages 2909–2916, 2020.

[Lukasiewicz *et al.*, 2022] Thomas Lukasiewicz, Enrico Malizia, Maria Vanina Martinez, Cristian Molinaro, Andreas Pieris, and Gerardo I. Simari. Inconsistency-tolerant query answering for existential rules. *Artif. Intell.*, 307:103685, 2022.

[Tsamoura *et al.*, 2021] Efthymia Tsamoura, David Carral, Enrico Malizia, and Jacopo Urbani. Materializing knowledge bases via trigger graphs. *Proc. VLDB Endow.*, 14(6):943–956, 2021.

[Vardi, 1982] Moshe Y. Vardi. The complexity of relational query languages. In *Proc. STOC*, pages 137–146, 1982.