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, Vienna University of Technology, Austria
[2]Department of Computer Science, University of Oxford, UK
[3]DISI, University of Bologna, Italy
[4]DIMES, University of Calabria, Italy

Abstract

Inconsistency-tolerant semantics provide meaningful answers to queries posed on possibly inconsistent knowledge bases. Recently, explainability has also become a prominent problem in different areas. 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 discuss recently introduced notions of explanations for this purpose along with a complexity analysis for a wide range of existential rule languages and for several complexity measures.

Keywords

Inconsistency-tolerant semantic, existential rules, explainability

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 [1] and generalized for several DLs [2]. Two other popular semantics introduced later on are the *intersection of repairs (IAR)* [2] and the *intersection of closed repairs (ICR)* [3] 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 [4], while for the *ICR* semantics, it has been observed in [5].

SEBD 2025: 33rd Symposium on Advanced Database Systems, June 16-19, 2025 - Ischia, Italy
*Corresponding author.
thomas.lukasiewicz@tuwien.ac.at (T. Lukasiewicz); enrico.malizia@unibo.it (E. Malizia); c.molinaro@dimes.unical.it (C. Molinaro)
0000-0002-6780-4711 (E. Malizia); 0000-0003-4103-1084 (C. Molinaro)

The above semantics' complexity is rather well-understood in the literature (see, e.g., [2, 3, 6, 7, 4, 5] for inconsistency-tolerant query answering in DLs, and, e.g., [8, 9, 10, 11, 12] 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 [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. [14, 15, 16] 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. Although DLs are popular ontology formalisms, rule-based ones are well-suited for data-intensive applications due to their favorable computational properties and expressiveness, especially when considering practical schema modeling, where higher-arity can often be handled through reification [24]. [22] considered explanations of positive query answers in the inconsistent setting under existential rules.

Explaining query answers under inconsistency-tolerant semantics includes the problem of explaining query *non*-entailment, which was studied for DLs in [16]. Here, we deal with the problem under existential rules, discussing the notions introduced in [25] along with the complexity analysis therein. Rather than showing which facts inside the database are in conflict with the entailment of the query, as in [16], 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. The complexity analysis that we discuss concern a wide spectrum of Datalog$^{\pm}$ languages under different complexity measures.

2. Preliminaries

In this section, we briefly recall some basics on existential rules from the context of Datalog$^{\pm}$ [26].

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 $h : \mathbf{C} \cup \mathbf{N} \cup \mathbf{V} \to \mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$ 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 of the following form: $\forall\mathbf{X}\forall\mathbf{Y}\,(\varphi(\mathbf{X}, \mathbf{Y}) \to \exists\mathbf{Z}\, p(\mathbf{X}, \mathbf{Z}))$, 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 $body(\sigma)$, while $p(\mathbf{X}, \mathbf{Z})$ is the *head* of σ, denoted $head(\sigma)$. 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 $\forall \mathbf{X}\,(\varphi(\mathbf{X}) \rightarrow \bot)$, where $\mathbf{X} \subseteq \mathbf{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms without nulls, called the *body* of ν and denoted $body(\nu)$, and $\bot$ denotes the truth constant *false*. An instance I satisfies ν, written $I \models \nu$, if there is *no* homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$. For 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. For a class of TGDs $\mathbb{C}$, $\mathbb{C}_\bot$ denotes 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 here considered guaranteeing decidability are among the most frequently analyzed in the literature: linear (L) [26], guarded (G) [27], sticky (S) [28], and acyclic TGDs (A), along with the "weak" (proper) generalizations weakly sticky (WS) [28] and weakly acyclic TGDs (WA) [29], 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. Recall that: $\mathsf{L} \subset \mathsf{G}$ and $\mathsf{F} \subset \mathsf{WA} \subset \mathsf{WS}$. A more detailed overview is in [12].

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., [27, 30], 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(L) the problem of BCQ answering when restricted over programs belonging to L.

Following Vardi [31], 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.

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 are the *ABox repair (AR)* semantics, its approximation by the *intersection of repairs (IAR)*, and the *intersection of closed repairs (ICR)* semantics [2, 3]; 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. Symmetrically, repairs have also been defined via *culprits*, which are inclusion-minimal subsets of D triggering an NC.

Repairs are hence obtained by deleting from D minimal hitting sets of all culprits [32, 33, 34].

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

We refer to [9, 12] 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.

Definition 3.1. *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.*

Example 3.2. *Consider the database $D = \{Prof(p), Reader(p), Chair(p), Dean(p), PVC(p)\}$, 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 TGDs Σ_T and the NCs Σ_{NC} reported below:*

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

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

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.

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

The next definition introduces (minimal) explanations for query entailment by a KB [18, 23].

Definition 3.3. *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 (minimal) explanations in [18], we require consistency, as here KBs can be inconsistent. The above concept of minimal explanations is equivalent to that of *causes* in [15, 16].

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 query MinExes (lost during the reparation). From this perspective, our definitions and the ones by Bienvenu et al. [16] 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 query MinExes; 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. [16] explanations provide facts F left in the repair R that are inconsistent with every query MinEx, whereas we provide facts E outside the repair R (deleted during the reparation) that intersect every query MinEx—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.

Below, we formally define our notions of explanations, which we aid by illustrating examples.

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 $\mathscr{E} = \{C\}$, 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 Theorem 3.2 and the query $q_1 = CanBeElectedFor(p, 3)$, 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 $\mathscr{E}_1 = \{C\}$, where $C = \{Reader(p), Chair(p)\}$. 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, and hence E does not appear in the intersection of the repairs.

Definition 3.6. *An* explanation for $KB \nvdash_{IAR} q$ *is a set* $\mathscr{E} = \{C_1, \ldots, C_n\}$*, with* $n \geq 1$*, where the* C_i*s 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 Theorem 3.2 and the query* $q_2 = \exists X\, CanBeElected(X)$*, asking whether there is someone who can be elected. Query* q_2 *is* not *entailed by KB under the IAR semantics. An explanation for* $KB \nvdash_{IAR} q_2$ *is* $\mathscr{E}_2 = \{C_1, C_2\}$*, where* $C_1 = \{Prof(p)\}$ *and* $C_2 = \{Reader(p)\}$*. 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 $KB \nvdash_{ICR} q$ 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, and hence E does not appear in the intersection of the repairs' closures. Given a finite set of facts E, q_E denotes the BCQ $\bigwedge_{f \in E} f$.

Definition 3.8. *An* explanation for $KB \nvdash_{ICR} q$ *is a set* $\mathscr{E} = \{C_1, \ldots, C_n\}$*, with* $n \geq 1$*, where the* C_i*s are subsets of repair deletions of KB such that, 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,* $X \cap C_i \neq \emptyset$*.*

Notice that $(Cl(KB), \Sigma)$ might be inconsistent, but every MinEx E for q w.r.t. $(Cl(KB), \Sigma)$, and 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. $(Cl(KB), \Sigma)$ not admitting any (consistent) MinEx for q_E w.r.t. KB; 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 Theorem 3.2's KB and the query* $q_3 = \exists X, Y\, CanBeElectedFor(X, Y)$*, 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* $E_1 = \{Prof(p)\}$*,* $E_2 = \{Reader(p), PVC(p)\}$*,* $E_3 = \{CanBeElectedFor(p, 1)\}$*, and* $E_4 = \{CanBeElectedFor(p, 3)\}$*. An explanation for* $KB \nvdash_{ICR} q_3$ *is* $\mathscr{E}_3 = \{C_1, C_2\}$*, where* $C_1 = \{Prof(p)\}$ *and* $C_2 = \{Reader(p), Chair(p)\}$*. 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* $X_1 = \{Prof(p), Chair(p)\}$ *and* $X_2 = \{Reader(p), PVC(p)\}$*, 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*.*

	Data	*fp*-comb.	*ba*-comb.	Comb.
$\mathsf{L}_\bot$, $\mathsf{LF}_\bot$, $\mathsf{AF}_\bot$	in P	D^{P}	$\mathrm{D}_2^{\mathrm{P}}$	PSPACE
$\mathsf{S}_\bot$, $\mathsf{SF}_\bot$	in P	D^{P}	$\mathrm{D}_2^{\mathrm{P}}$	EXP
$\mathsf{A}_\bot$	in P	D^{P}	$\mathrm{P}^{\mathrm{NEXP}}$	$\mathrm{P}^{\mathrm{NEXP}}$
$\mathsf{G}_\bot$	D^{P}	D^{P}	EXP	2EXP
$\mathsf{F}_\bot$, $\mathsf{GF}_\bot$	D^{P}	D^{P}	$\mathrm{D}_2^{\mathrm{P}}$	EXP
$\mathsf{WS}_\bot$, $\mathsf{WA}_\bot$	D^{P}	D^{P}	2EXP	2EXP

Table 1
Complexity of *AR*-/*IAR*-/*ICR*-MinEx$^{\nvdash}$. All entries without "in" are completeness results. All these results hold also for deciding whether a set is a (not necessarily minimal) explanation.

Definition 3.10. *For each $S \in \{AR, IAR, ICR\}$, an explanation $\mathscr{E} = \{C_1, \ldots, C_n\}$ for $KB \nvdash_S q$ is* minimal *iff there is no explanation for $KB \nvdash_S q$ that can be derived from $\mathscr{E}$ 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 $\mathscr{E}$. Notice that if an explanation $\mathscr{E}$ contains a "redundant" C_i, that is, $\mathscr{E} \setminus \{C_i\}$ is an explanation, then $\mathscr{E}$ is not minimal according to the definition above, as we can delete all facts from C_i and still get an explanation, that is, $\mathscr{E}' = \mathscr{E} \setminus \{C_i\} \cup \{\emptyset\}$ is an explanation. In fact, even $\mathscr{E}'$ is not minimal, as the empty set can be merged with any other C_j in $\mathscr{E}'$, yielding the (possibly minimal) explanation $\mathscr{E} \setminus \{C_i\}$.

In this paper, we focus on the following decision problems.

Problem: S-MinEx$^{\nvdash}(L)$, with $S \in \{AR, IAR, ICR\}$.
Input: A knowledge base $KB = (D, \Sigma)$ with $\Sigma \in L$, a BCQ q, and $\mathscr{E} \subseteq P(D)$, with $P(D)$ being the powerset of D.
Question: Is $\mathscr{E}$ a minimal explanation for $KB \nvdash_S q$?

4. Complexity Analysis

Our complexity results are summarized in Table 1. The complexity ranges from membership in P to 2EXP-completeness. For details on how the results have been derived, we refer the reader to [25]. Here we focus on the main takeaways the complexity analysis provides.

First, notice that the three inconsistency-tolerant semantics have the same complexity across all languages and all complexity measures considered in this paper. This is not the case when explaining positive query answers (cf. [22]): in such a setting, the *AR* and *ICR* have the same complexity, which is in turn at least as high as the one of the *IAR* semantics.

In the following, we compare explanations for negative and positive query answers (both under inconsistency-tolerant semantics) in terms of the complexity of deciding whether a given set of sets of facts is a minimal explanation, naming the two settings "negative" and "positive".

The complexity for the *AR* and *ICR* semantics does not change under the two settings except for the data complexity of first-order rewritable languages (namely, $\mathsf{L}_\bot$, $\mathsf{S}_\bot$, $\mathsf{A}_\bot$, and their sub-languages), which is in P for the negative setting and is D^{P}-complete for the positive one. Thus, the complexity of the positive setting is at least as high as the one of the negative setting.

As for the *IAR* semantics, the complexity does not change under the two settings except for *(i)* the data complexity of $\mathsf{L}_\bot$ and its generalizations, which is co-NP-complete for the positive setting and D^P-complete for the negative one; *(ii)* the bounded-arity combined complexity of $\mathsf{F}_\bot$, $\mathsf{L}_\bot$, $\mathsf{S}_\bot$, and their specializations, which is Π_2^P-complete for the positive setting and D_2^P-complete for the negative one. Thus, in contrast to the *AR* and *ICR* semantics, for the *IAR* semantics the complexity might increase when moving from the positive to the negative setting.

5. Summary and Outlook

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 [15, 16] for DL-Lite$_\mathscr{R}$. 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. Or considering preferences over repairs, similar to what is done in [10, 11], by also introducing more elaborate models [35, 36, 37, 38], where constraints and compact representations can be considered [39, 40, 41, 42, 43, 44].

Acknowledgments

Enrico Malizia's work was supported by the European Union – NextGenerationEU programme, through the Italian Ministry of University and Research (MUR) PRIN 2022-PNRR grant P2022KHTX7 "DISTORT"—CUP: J53D23015000001, under the Italian "National Recovery and Resilience Plan" (PNRR), Mission 4 Component 1.

Cristian Molinaro's work was supported by the Italian Ministry of University and Research (MUR) PRIN 2022 grant 2022XERWK9 "S-PIC4CHU - Semantics-based Provenance, Integrity, and Curation for Consistent, High-quality, and Unbiased data science"— CUP: H53C24000990006.

Cristian Molinaro acknowledges the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Arenas, L. E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: Proc. PODS, 1999, pp. 68–79.

[2] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, D. F. Savo, Inconsistency-tolerant semantics for description logics, in: Proc. RR, 2010, pp. 103–117.
[3] M. Bienvenu, On the complexity of consistent query answering in the presence of simple ontologies, in: Proc. AAAI, 2012, pp. 705–711.
[4] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, D. F. Savo, Inconsistency-tolerant query answering in ontology-based data access, J. Web Sem. 33 (2015) 3–29.
[5] M. Bienvenu, C. Bourgaux, Inconsistency-tolerant querying of description logic knowledge bases, in: Reasoning Web, 2016, pp. 156–202.
[6] M. Bienvenu, R. Rosati, Tractable approximations of consistent query answering for robust ontology-based data access, in: Proc. IJCAI, 2013, pp. 775–781.
[7] M. Bienvenu, C. Bourgaux, F. Goasdoué, Querying inconsistent description logic knowledge bases under preferred repair semantics, in: Proc. AAAI, 2014, pp. 996–1002.
[8] T. Eiter, T. Lukasiewicz, L. Predoiu, Generalized consistent query answering under existential rules, in: Proc. KR, 2016, pp. 359–368.
[9] T. Lukasiewicz, E. Malizia, C. Molinaro, Complexity of approximate query answering under inconsistency in Datalog+/−, in: Proc. IJCAI, 2018, pp. 1921–1927.
[10] 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.
[11] T. Lukasiewicz, E. Malizia, C. Molinaro, Complexity of inconsistency-tolerant query answering in datalog+/− under preferred repairs, in: Proc. KR, 2023, pp. 472–481.
[12] 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) 103685.
[13] A. Arioua, N. Tamani, M. Croitoru, Query answering explanation in inconsistent Datalog+/− knowledge bases, in: Proc. DEXA, 2015, pp. 203–219.
[14] M. Bienvenu, C. Bourgaux, F. Goasdoué, Explaining query answers under inconsistency-tolerant semantics over description logic knowledge bases, in: Proc. DL, 2015.
[15] M. Bienvenu, C. Bourgaux, F. Goasdoué, Explaining inconsistency-tolerant query answering over description logic knowledge bases, in: Proc. AAAI, 2016, pp. 900–906.
[16] M. Bienvenu, C. Bourgaux, F. Goasdoué, Computing and explaining query answers over inconsistent DL-Lite knowledge bases, J. Artif. Intell. Res. 64 (2019) 563–644.
[17] A. Hecham, A. Arioua, G. Stapleton, M. Croitoru, An empirical evaluation of argumentation in explaining inconsistency-tolerant query answering, in: Proc. DL, 2017.
[18] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Explanations for query answers under existential rules, in: Proc. IJCAI, 2019, pp. 1639–1646.
[19] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicenavicius, Explanations for negative query answers under existential rules, in: Proc. KR, 2020, pp. 223–232.
[20] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavicius, Explanations for ontology-mediated query answering in description logics, in: Proc. ECAI, 2020, pp. 672–679.
[21] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicenavicius, Preferred explanations for ontology-mediated queries under existential rules, in: Proc. AAAI, 2021, pp. 6262–6270.
[22] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for inconsistency-tolerant query answering under existential rules, in: Proc. AAAI, 2020, pp. 2909–2916.
[23] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Explanations for query answers

under existential rules, Artif. Intell. 341 (2025) 104294.
[24] A. Calì, D. Calvanese, G. D. Giacomo, M. Lenzerini, A formal framework for reasoning on UML class diagrams, in: Proc. ISMIS, volume 2366, Springer, 2002, pp. 503–513.
[25] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for negative query answers under inconsistency-tolerant semantics, in: Proc. IJCAI, 2022, pp. 2705–2711.
[26] A. Calì, G. Gottlob, T. Lukasiewicz, A general Datalog-based framework for tractable query answering over ontologies, J. Web Sem. 14 (2012) 57–83.
[27] A. Calì, G. Gottlob, M. Kifer, Taming the infinite chase: Query answering under expressive relational constraints, J. Artif. Intell. Res. 48 (2013) 115–174.
[28] A. Calì, G. Gottlob, A. Pieris, Towards more expressive ontology languages: The query answering problem, Artif. Intell. 193 (2012) 87–128.
[29] R. Fagin, P. G. Kolaitis, R. J. Miller, L. Popa, Data exchange: semantics and query answering, Theor. Comput. Sci. 336 (2005) 89–124.
[30] E. Tsamoura, D. Carral, E. Malizia, J. Urbani, Materializing knowledge bases via trigger graphs, Proc. VLDB Endow. 14 (2021) 943–956.
[31] M. Y. Vardi, The complexity of relational query languages, in: Proc. STOC, 1982, pp. 137–146.
[32] J. Chomicki, J. Marcinkowski, Minimal-change integrity maintenance using tuple deletions, Inf. Comput. 197 (2005) 90–121.
[33] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, in: Proc. LICS, 2014, pp. 43:1–43:10.
[34] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, SIAM J. Comput. 47 (2018) 456–492.
[35] T. Lukasiewicz, E. Malizia, On the complexity of *m*CP-nets, in: Proc. AAAI, 2016, pp. 558–564.
[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. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On preferences and priority rules in abstract argumentation, in: Proc. IJCAI, 2022, pp. 2517–2524.
[39] 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.
[40] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Preference-based inconsistency-tolerant query answering under existential rules, Artif. Intell. 312 (2022) 103772.
[41] M. Calautti, L. Caroprese, S. Greco, C. Molinaro, I. Trubitsyna, E. Zumpano, Existential active integrity constraints, Expert Syst. Appl. 168 (2021) 114297.
[42] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, The complexity of the nucleolus in compact games, ACM Trans. Comput. Theory 7 (2014) 3:1–3:52.
[43] E. Malizia, L. Palopoli, F. Scarcello, Infeasibility certificates and the complexity of the core in coalitional games, in: Proc. IJCAI, 2007, pp. 1402–1407.
[44] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, On the complexity of compact coalitional games, in: Proc. IJCAI, 2009, pp. 147–152.