

Explanations for Inconsistency-Tolerant Query Answering under Existential Rules^{*}

(Discussion Paper)

Thomas Lukasiewicz¹, Enrico Malizia² and Cristian Molinaro³

¹Department of Computer Science, University of Oxford, UK

²DISI, University of Bologna, Italy

³DIMES, University of Calabria, Italy

Abstract

Querying inconsistent knowledge bases has attracted a great deal of interest over the last decades. Also explainability has recently become a prominent problem in AI, and explaining query answers allows users to understand why a query is entailed. In this paper, we address the problem of explaining ontological query answers in the existential rules setting under three popular inconsistency-tolerant semantics, namely, the ABox repair, the intersection of repairs, and the intersection of closed repairs semantics.

Keywords

Knowledge representation, Existential rules, Inconsistencies, Query answering, Explanations, Complexity

1. Introduction

Existential rules from the context of Datalog[±] [2] and description logics (DLs) [3] are popular ontology languages. In real-world applications, it may very well be the case that the data are inconsistent with the ontology. To provide meaningful answers to queries in the presence of inconsistency, various inconsistency-tolerant semantics of query answering have been proposed.

In the *ABox repair* (*AR*) semantics, first developed for relational databases [4] and then generalized for several DLs [5], a query answer is valid if it can be inferred from each of the *repairs* of the knowledge base, that is, the inclusion-maximal consistent subsets of the database. The *intersection of repairs* (*IAR*) [5] and the *intersection of closed repairs* (*ICR*) [6] semantics have been introduced as approximations of the *AR* semantics (see also [7, 8, 9] for other approximation approaches). An answer is considered to be valid under the *IAR* (resp., *ICR*) semantics if it can be inferred from the intersection of the repairs (resp., the intersection of the closure of the repairs), along with the ontology.

Explainability has recently become a prominent problem in different areas of AI. In our setting, explaining query answers allows users to understand not only what is entailed by an


SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy


^{*}This is a short version of the AAAI-20 paper [1].

✉ thomas.lukasiewicz@cs.ox.ac.uk (T. Lukasiewicz); enrico.malizia@unibo.it (E. Malizia);

cmolinaro@dimes.unical.it (C. Molinaro)

ORCID 0000-0002-7644-1668 (T. Lukasiewicz); 0000-0002-6780-4711 (E. Malizia); 0000-0003-4103-1084 (C. Molinaro)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

inconsistent knowledge base, but also *why* it is entailed. In this paper, we study explanations of query entailment under inconsistency-tolerant semantics in the presence of existential rules.P

There have been various works on explanations for query answers under existential rules in the *consistent* setting. Explaining query answers under existential rules was investigated in [10] and under DL in [11]; preferred explanations in [12] and negative explanations in [13].

Explaining query answers under inconsistency-tolerant semantics has recently been addressed in the literature. Arioua et al. [14] addressed the problem of explaining query entailment under the ICR semantics in the presence of existential rules for which the Skolemized chase is finite. Their definition of explanation is based on abstract argumentation. Their approach along with interactive explanation methods based on dialectical approaches has been experimentally evaluated by Hecham et al. [15]. Bienvenu et al. [16, 17, 18] considered the logic DL-Lite_R. They defined explanations for positive and negative answers under the brave, *AR*, and *IAR* semantics, and investigated the data complexity of different related problems.

In this paper we investigate the complexity of query explanations under the *AR*, *IAR*, and *ICR* semantics for a wide spectrum of Datalog[±] languages.

2. Preliminaries

We here briefly recall some basics on existential rules from the context of Datalog[±] [2].

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 $p(t_1, \dots, t_n)$, where p is an n -ary predicate, and t_1, \dots, 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 $h: \mathbf{C} \cup \mathbf{N} \cup \mathbf{V} \mapsto \mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$ 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 $|\mathbf{X}|$ -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; 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. 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 $\forall \mathbf{X} \varphi(\mathbf{X}) \rightarrow \perp$, where $\mathbf{X} \subseteq \mathbf{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms without nulls, and \perp 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.

Table 1

Complexity of BCQ answering under existential rules [22]. All non-“in” entries are completeness results.

\mathcal{L}	Data	<i>fp-comb.</i>	<i>ba-comb.</i>	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 a TGD class \mathbb{C} , \mathbb{C}_\perp 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 \mathcal{L} that we consider to guarantee decidability are among the most frequently analyzed in the literature, namely, linear (L) [2], guarded (G) [19], sticky (S) [20], and acyclic TGDs (A), along with the “weak” (proper) generalizations weakly sticky (WS) [20] and weakly acyclic TGDs (WA) [21], 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 [22] 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 TGDs and NCs subsets, respectively, of Σ . The set $mods(KB)$ of *models* of $KB = (D, \Sigma)$ is the set of instances $\{I \mid I \supseteq D \wedge I \models \Sigma\}$; KB is *consistent* if $mods(KB) \neq \emptyset$, otherwise KB is *inconsistent*. The *answer* to a CQ q w.r.t. 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 $ans(q, KB) \neq \emptyset$. Another way to define the existential rules semantics is via the concept of the *Chase* (see, e.g., [23, 24]). The decision version of the *CQ answering* problem is: for 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. $BCQ(\mathcal{L})$ denotes the problem of BCQ answering when restricted over programs belonging to \mathcal{L} .

Following Vardi [25], 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 the complexity results of BCQ answering for the languages here considered [22].

A language \mathcal{L} is *FO-rewritable* if given any program $\Sigma \in \mathcal{L}$ and any BCQ q , there exists an FO-query q_Σ such that, for all databases D we have that $(D, \Sigma) \models q$ iff $D \models q_\Sigma$. All languages from Table 1 with AC^0 data complexity are FO-rewritable.

Inconsistency-Tolerant Semantics. In classical BCQ answering, for an inconsistent knowledge base KB (i.e., $mods(KB) = \emptyset$), every query is entailed, as everything follows from a contradiction. Clearly, the answers obtained in such cases are not meaningful. Three prominent

inconsistency-tolerant semantics for 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 [5, 6]; all three are based on the notion of *repair*.

A *repair* of a knowledge base $KB = (D, \Sigma)$ is an inclusion-maximal subset R of D such that $\text{mods}((R, \Sigma)) \neq \emptyset$; $\text{Rep}(KB)$ is the set of all KB 's repairs. The *closure* $Cl(KB)$ of KB 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, if $(R, \Sigma) \models q$ for all $R \in \text{Rep}(KB)$.
- KB entails q under the *intersection of repairs* (*IAR*) semantics, if $(D_I, \Sigma) \models q$, where $D_I = \bigcap \{R \mid R \in \text{Rep}(KB)\}$.
- KB entails q under the *intersection of closed repairs* (*ICR*) semantics, if $(D_C, \Sigma) \models q$, where $D_C = \bigcap \{Cl(R, \Sigma) \mid R \in \text{Rep}(KB)\}$.

Symmetrically, the concept of repair is linked to that of culprit. Intuitively, a culprit is a minimal subset of D that, together with Σ_T entails some NC; a culprit for an NC is a “minimal explanation” [10] (see below) of the NC. By deleting from D a minimal hitting set [26, 27] of facts S intersecting all culprits, we obtain a repair $R = D \setminus S$.

We refer to [22, 28, 29] for more on the complexity of *AR*-/*IAR*-/*ICR*-query answering.

3. Explanations for Query Answers

An *explanation* for q w.r.t. KB is a subset E of D such that (E, Σ) is consistent and $(E, \Sigma) \models q$. 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 . We now introduce the notions of (minimal) explanation under the *AR*, *IAR*, and *ICR* semantics.

- Definition 1.** • An *AR-explanation* for q w.r.t. KB is a set of explanations $\mathcal{E} = \{E_1, \dots, E_n\}$ for q w.r.t. KB such that every repair of KB contains some E_i .
- An *IAR-explanation* for q w.r.t. KB is a singleton set of explanations $\mathcal{E} = \{E\}$ for q w.r.t. KB such that $E \subseteq R$ for every repair $R \in \text{Rep}(KB)$.
 - An *ICR-explanation* for q w.r.t. KB is a set of explanations $\mathcal{E} = \{E_1, \dots, E_n\}$ for q w.r.t. KB such that each KB 's repair contains an E_i and $(E_C, \Sigma) \models q$, with $E_C = \bigcap \{Cl(E_i) \mid E_i \in \mathcal{E}\}$.

Definition 2. For any $S \in \{AR, IAR, ICR\}$, an S -explanation $\mathcal{E} = \{E_1, \dots, E_n\}$ for q w.r.t. KB is an S -minimal explanation, or S -MinEx, if every $E_i \in \mathcal{E}$ is a MinEx for q w.r.t. KB , and no $\mathcal{E}' \subsetneq \mathcal{E}$ is an S -explanation for q w.r.t. KB .

Example 3. Consider the database $D = \{Prof(p, cs), Postdoc(p, math), Group(g)\}$, asserting that p is a professor working in the *cs* department, p is a postdoc working in the *math* department, and g is a research group. Consider also the following program Σ :

$$\begin{aligned} Prof(X, Y) &\rightarrow Researcher(X), & Prof(X, Y) &\rightarrow Dept(Y), \\ Postdoc(X, Y) &\rightarrow Researcher(X), & Postdoc(X, Y) &\rightarrow Dept(Y), \\ Prof(X, Y), Postdoc(X, Z) &\rightarrow \perp, \end{aligned}$$

expressing that *Prof* and *Postdoc* have *Researcher* as domain and *Dept* as range, and one cannot be both a professor and a postdoc. Clearly, the knowledge base $KB = (D, \Sigma)$ is

Table 2

Complexity of *AR*-, *IAR*-, and *ICR*-MinEx. All entries are completeness results. Hardness results for *AR* and *ICR* in the data and *fp*-combined complexity also follow from inspection of a proof in [18].

\mathcal{L}	<i>AR</i> - and <i>ICR</i> -MinEx				<i>IAR</i> -MinEx			
	Data	<i>fp</i> -comb.	<i>ba</i> -comb.	Comb.	Data	<i>fp</i> -comb.	<i>ba</i> -comb.	Comb.
$L_{\perp}, LF_{\perp}, AF_{\perp}$	D^P	D^P	D_2^P	PSPACE	in P	D^P	Π_2^P	PSPACE
S_{\perp}, SF_{\perp}	D^P	D^P	D_2^P	EXP	in P	D^P	Π_2^P	EXP
A_{\perp}	D^P	D^P	P^{NEXP}	P^{NEXP}	in P	D^P	P^{NEXP}	P^{NEXP}
G_{\perp}	D^P	D^P	EXP	2EXP	co-NP	D^P	EXP	2EXP
F_{\perp}, GF_{\perp}	D^P	D^P	D_2^P	EXP	co-NP	D^P	Π_2^P	EXP
WS_{\perp}, WA_{\perp}	D^P	D^P	2EXP	2EXP	co-NP	D^P	2EXP	2EXP

inconsistent, as p violates the NC. The knowledge base admits the following two repairs: $D' = \{Prof(p, cs), Group(g)\}$, and $D'' = \{Postdoc(p, math), Group(g)\}$. Their intersection is $D_I = \{Group(g)\}$, while their closures' intersection is $D_C = \{Group(g), Researcher(p)\}$.

The BCQ $\exists X Group(X)$ is entailed by KB under *IAR* (and thus also under *ICR* and *AR*). The set $\{\{Group(g)\}\}$ is an *IAR*-minimal (as well as *ICR*- and *AR*-minimal) explanation for the query w.r.t. KB . Indeed, $Group(g)$ is the fact in D_I that entails the query.

The BCQ $\exists X Researcher(X)$ is entailed by KB under *ICR* (and thus also under *AR*), but not under *IAR*. The set $\{\{Prof(p, cs)\}, \{Postdoc(p, math)\}\}$ is an *ICR*-minimal (as well as *AR*-minimal) explanation for the query w.r.t. KB . Indeed, $Researcher(p)$ is the fact in D_C that entails the query, and the reason why $Researcher(p)$ belongs to the closures of D' and D'' are the facts $Prof(p, cs)$ and $Postdoc(p, math)$ of D' and D'' , respectively.

The BCQ $\exists X Dept(X)$ is entailed by KB only under *AR*. An *AR*-minimal explanation for the query w.r.t. KB is $\{\{Prof(p, cs)\}, \{Postdoc(p, math)\}\}$. Indeed, $Prof(p, cs)$ is the fact of D' entailing the query, while $Postdoc(p, math)$ is the fact of D'' entailing the query. \triangleleft

We will study the *S*-MinEx problems, for any $S \in \{AR, IAR, ICR\}$, of recognizing *S*-MinExes: for a knowledge base $KB = (D, \Sigma)$, a BCQ q , and a set $\mathcal{E} \subseteq \mathcal{P}(D)$, with $\mathcal{P}(D)$ being the powerset of D , decide whether the set \mathcal{E} is an *S*-MinEx for q w.r.t. KB .

4. Complexity Analysis

The first results imply most of the complexity upper-bounds of Table 2. The intuition behind these theorems is as follows (for the details see [1]). Verifying that \mathcal{E} is an *S*-MinEx for q w.r.t. KB requires checking the following conditions: (1) that all $E_i \in \mathcal{E}$ are MinExes of q w.r.t. KB (which implies verifying that: (1a) all $E_i \in \mathcal{E}$ are consistent; (1b) all $E_i \in \mathcal{E}$ entail q ; and (1c) all $E_i \in \mathcal{E}$ are minimal); (2) that all the repairs are “covered” by \mathcal{E} ; and (3) verify that the “cover by \mathcal{E} is minimal”, for which we borrow the concept of “critical vertex in a minimal hitting set” [30].

Theorem 4. *Let \mathcal{L} be one of the Datalog[±] languages of this paper. If BCQ(\mathcal{L}) is in \mathbf{C} , then *AR*-MinEx and *IAR*-MinEx can be answered by the following sequence of checks:*

- *AR*: a co-(NP^C) check, an NP^C check, a linear number of \mathbf{C} checks, and a linear number of co- \mathbf{C} checks.

- *IAR*: a $\text{co-}(\text{NP}^{\text{C}})$ check, a **C** check, and a linear number of $\text{co-}\mathbf{C}$ checks.

For the *ICR* case we need a slightly different proof. Verifying that a set $\mathcal{E} = \{E_1, \dots, E_n\}$ is an *ICR-MinEx* requires to check conditions (1), (2), and (3) as above, and the additional condition that the intersection of the closure of all the E_i 's entails the query. Verifying the latter can be reduced to *ICR* reasoning over a suitable knowledge base.

Theorem 5. *Let \mathcal{L} be one of the Datalog^{\pm} languages of this paper. If $\text{BCQ}(\mathcal{L})$ (resp., $\text{BCQ}(\mathcal{L})$ under *ICR*) is in **C** (resp., in **D**), then *ICR-MinEx* can be answered by the following sequence of checks: a $\text{co-}(\text{NP}^{\text{C}})$ check, an NP^{C} check, a **D** check, and a linear number of $\text{co-}\mathbf{C}$ checks.*

For the *fp*-combined setting we have tighter results thanks to these two observations. First, in the *fp*-combined setting, for the Datalog^{\pm} languages here considered, checking a set of facts to be a repair is in **P**. Second, for the *ICR* case, we also need to notice that, in the *fp*-combined setting, checking the intersection of the E_i 's closure to entail the query is in **NP**.

Theorem 6. *AR -/ IAR -/ ICR -*MinEx* is in D^{P} in the *fp*-combined complexity for the Datalog^{\pm} languages of this paper.*

The following result proves the **P** upper-bounds in Table 2. A key observation is that the intersection of the repairs in the stated fragments is computable in polynomial time [22].

Theorem 7. *IAR -*MinEx* from knowledge bases over L_{\perp} , A_{\perp} , and S_{\perp} is in **P** in the data complexity.*

The upper-bounds found in the previous section are actually tight, and indeed we can show matching lower-bounds. The co-NP -hardness and Π_2^{P} -hardness results for *IAR* are via reductions from UNSAT and from deciding the validity of a QBF $\forall X \exists Y \phi(X, Y)$, respectively.

The P^{NEXP} -hardness results are obtained via a reduction from the following problem [22]: 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 .

The D^{P} -hardness results in the data complexity for *AR* and *ICR* are via a reduction from MINIMAL UNSAT [31]: given a Boolean formula ϕ , decide whether ϕ is *minimally unsatisfiable*, i.e., if ϕ is unsatisfiable, *and* removing any clause from ϕ makes the formula satisfiable.

The D_2^{P} -hardness results for *AR* and *ICR* are via a reduction from the problem of deciding the validity of two QBFs $\Phi = \exists X \forall Y \neg \phi(X, Y)$ and $\Psi = \forall X \exists Y \psi(X, Y)$. The reduction uses the simplifying assumption that variables X and Y of Φ and Ψ can be the same [32, 33].

The remaining hardness results follow from IS-MINEX 's hardness over consistent KBs [10].

5. Summary and Outlook

We have analyzed the problem of explaining query answers under three popular inconsistency-tolerant semantics, for a wide range of existential rules, and under different complexity measures; this work has recently been extended to negative query answers [34].

This paper opens up several avenues for further research, like analyzing the complexity of other related problems, such as deciding if a fact is *necessary* or *relevant*. Inspired by the

idea of exploring preferences over explanations [12], we can also consider how more elaborate preference models can be included in this framework [35, 36, 37, 38, 39]. Moreover, in some scenarios, knowing the existential rules needed to derive query answers may be useful as well.

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] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for inconsistency-tolerant query answering under existential rules, in: *AAAI*, 2020, pp. 2909–2916.
- [2] A. Cali, G. Gottlob, T. Lukasiewicz, A general Datalog-based framework for tractable query answering over ontologies, *J. Web Sem.* 14 (2012) 57–83.
- [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] M. Arenas, L. E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: *PODS*, 1999, pp. 68–79.
- [5] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, D. F. Savo, Inconsistency-tolerant semantics for description logics, in: *RR*, 2010, pp. 103–117.
- [6] M. Bienvenu, On the complexity of consistent query answering in the presence of simple ontologies, in: *AAAI*, 2012, pp. 705–711.
- [7] S. Greco, C. Molinaro, Probabilistic query answering over inconsistent databases, *Ann. Math. Artif. Intell.* 64 (2012) 185–207.
- [8] S. Greco, C. Molinaro, I. Trubitsyna, Computing approximate query answers over inconsistent knowledge bases, in: *IJCAI*, 2018, pp. 1838–1846.
- [9] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Preference-based inconsistency-tolerant query answering under existential rules, in: *KR*, 2020, pp. 203–212.
- [10] Í. Í. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenaivičius, Explanations for query answers under existential rules, in: *IJCAI*, 2019, pp. 1639–1646.
- [11] Í. Í. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenaivičius, Explanations for ontology-mediated query answering in description logics, in: *ECAI*, 2020, pp. 672–679.
- [12] Í. Í. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicenaivičius, Preferred explanations for ontology-mediated queries under existential rules, in: *AAAI*, 2021, pp. 6262–6270.
- [13] Í. Í. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicenaivičius, Explanations for negative query answers under existential rules, in: *KR*, 2020, pp. 223–232.
- [14] A. Arioua, N. Tamani, M. Croitoru, Query answering explanation in inconsistent Datalog+/-knowledge bases, in: *DEXA*, 2015, pp. 203–219.
- [15] A. Hecham, A. Arioua, G. Stapleton, M. Croitoru, An empirical evaluation of argumentation in explaining inconsistency-tolerant query answering, in: *DL*, 2017.
- [16] M. Bienvenu, C. Bourgaux, F. Goasdoué, Explaining query answers under inconsistency-tolerant semantics over description logic knowledge bases, in: *DL*, 2015.

- [17] M. Bienvenu, C. Bourgaux, F. Goasdoué, Explaining inconsistency-tolerant query answering over description logic knowledge bases, in: AAAI, 2016, pp. 900–906.
- [18] 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.
- [19] A. Cali, G. Gottlob, M. Kifer, Taming the infinite chase: Query answering under expressive relational constraints, *J. Artif. Intell. Res.* 48 (2013) 115–174.
- [20] A. Cali, G. Gottlob, A. Pieris, Towards more expressive ontology languages: The query answering problem, *Artif. Intell.* 193 (2012) 87–128.
- [21] R. Fagin, P. G. Kolaitis, R. J. Miller, L. Popa, Data exchange: Semantics and query answering, *Theor. Comput. Sci.* 336 (2005) 89–124.
- [22] 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.
- [23] E. Tsamoura, D. Carral, E. Malizia, J. Urbani, Materializing knowledge bases via trigger graphs, *VLDB Endow.* 14 (2021) 943–956.
- [24] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Exploiting equality generating dependencies in checking chase termination, *VLDB Endow.* 9 (2016) 396–407.
- [25] M. Vardi, The complexity of relational query languages, in: STOC, 1982, pp. 137–146.
- [26] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, in: LICS, 2014, pp. 43:1–43:10.
- [27] J. Chomicki, J. Marcinkowski, Minimal-change integrity maintenance using tuple deletions, *Inf. Comput.* 197 (2005) 90–121.
- [28] T. Lukasiewicz, E. Malizia, A. Vaiceničius, Complexity of inconsistency-tolerant query answering in Datalog+/- under cardinality-based repairs, in: AAAI, 2019, pp. 2962–2969.
- [29] T. Lukasiewicz, E. Malizia, C. Molinaro, Complexity of approximate query answering under inconsistency in Datalog[±], in: IJCAI, 2018, pp. 1921–1927.
- [30] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, *SIAM J. Comput.* 47 (2018) 456–492.
- [31] C. H. Papadimitriou, D. Wolfe, The complexity of facets resolved, *J. Comput. Syst. Sci.* 37 (1988) 2–13.
- [32] T. Lukasiewicz, E. Malizia, On the complexity of m CP-nets, in: AAAI, 2016, pp. 558–564.
- [33] 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.
- [34] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for negative query answers under inconsistency-tolerant semantics, in: IJCAI, 2022.
- [35] 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.
- [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] M. Calautti, L. Caroprese, S. Greco, C. Molinaro, I. Trubitsyna, E. Zumpano, Existential active integrity constraints, *Expert Syst. Appl.* 168 (2021) art. no. 114297.
- [39] 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.