Query Answer Explanations under Existential Rules*

(Discussion Paper)

İsmail İlkan Ceylan[1], Thomas Lukasiewicz[1], Enrico Malizia[2] and
Andrius Vaicenavičius[1]

[1] *Department of Computer Science, University of Oxford, UK*
[2] *DISI, University of Bologna, Italy*

Abstract

Ontology-mediated query answering is an extensively studied paradigm, which aims at improving query answers with the use of a logical theory. In this paper, we focus on ontology languages based on existential rules, and we carry out a thorough complexity analysis of the problem of explaining query answers in terms of minimal subsets of database facts and related tasks.

Keywords

Knowledge representation, Existential rules, Query answering, Explanations, Computational complexity

1. Introduction

Ontology-based data access (OBDA) is a popular paradigm in knowledge representation and reasoning [2]. In this paradigm, an *ontology* enriches the user query with commonsense knowledge, to form a so-called *ontology-mediated query (OMQ)* [3]. The task of evaluating OMQs is called *ontology-mediated query answering (OMQA)*. *Description logics (DLs)* [4] and *existential rules* [5] are the most widely used ontology languages. Tracking down the causes of entailment in DL ontologies has attracted extensive interest. In the DL context, a prominent approach is identifying explanations as ontology axioms subsets [6, 7, 8], which are called *justifications* [9, 10]. Axiom pinpointing is extensively studied in DLs, and some implementations exist as well [6, 11]. However, most of the existing approaches to explanations in the DL framework focus on classical reasoning tasks. Explaining query entailments has only been investigated for the *DL-Lite* family of languages [12, 13], and *negative* answers to OMQs in DLs [14].

In this paper, we study explanations in the existential rules context. The only prior work related to explanations in existential rules is given in [15] in the context of *probabilistic databases*, and hence of a very different flavor. Here, given an OMQ, we are interested in explaining the OMQ in terms of the minimal satisfying subsets, called *minimal explanations* of a given database.

We analyze the complexity for each of the problems introduced in terms of *data*, *fp-combined*, *ba-combined* and *combined* complexity, for a representative set of existential rules.

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

* This is a short version of the IJCAI-19 paper [1].

ismail.ceylan@cs.ox.ac.uk (İ. İ. Ceylan); thomas.lukasiewicz@cs.ox.ac.uk (T. Lukasiewicz);
enrico.malizia@unibo.it (E. Malizia); andrius.vaicenavicius@cs.ox.ac.uk (A. Vaicenavičius)
0000-0003-4118-4689 (İ. İ. Ceylan); 0000-0002-7644-1668 (T. Lukasiewicz); 0000-0002-6780-4711 (E. Malizia)

CEUR Workshop Proceedings (CEUR-WS.org)

2. Preliminaries

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

General. We assume a set $\mathbf{C}$ of *constants*, a set $\mathbf{N}$ of *labeled nulls*, and a set $\mathbf{V}$ of *variables*. A *term* t is a constant, null, or variable. We assume a set of *predicates*, each associated with an arity. An *atom* has the form $p(t_1, \ldots, t_n)$, where p is an n-ary predicate, and $t_1, \ldots, t_n$ are terms. An atom containing only constants is called *fact*. Conjunctions of atoms are also identified with the sets of their atoms. An *instance* I is a (possibly infinite) set of atoms defined over constants and nulls. A *database* D is a finite instance containing only constants. A *homomorphism* is a substitution $h\colon \mathbf{C} \cup \mathbf{N} \cup \mathbf{V} \mapsto \mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$ 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 $h'(p(\mathbf{X}, \mathbf{Z})) \in I$. Given a set Σ of TGDs, I satisfies Σ, written $I \models \Sigma$, if I satisfies each TGD of Σ. For brevity, we omit the universal quantifiers in front of TGDs, and use the comma (instead of $\wedge$) for conjoining atoms. Finite sets of TGDs are called *programs* (or, *ontologies*), 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) [5], guarded (G) [16], sticky (S) [17], and acyclic TGDs (A), along with the "weak" (proper) generalizations weakly sticky (WS) [17] and weakly acyclic TGDs (WA) [18], 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: $\mathsf{L} \subset \mathsf{G}$ and $\mathsf{F} \subset \mathsf{WA} \subset \mathsf{WS}$. We refer to [19] for a more detailed overview.

Ontology-Mediated Query Answering. The OMQA paradigm generalizes query answering over databases by incorporating additional background knowledge via an ontology.

Formally, an *ontology-mediated query (OMQ)* is a pair (q, Σ), where q is a query, and Σ is an ontology. Let $\mathcal{L}$ be a Datalog$^{\pm}$ language. If $\Sigma \in \mathcal{L}$, then we say that (q, Σ) is an $\mathcal{L}$-OMQ. Consider a database D and an OMQ (q, Σ). The set of *models* of (D, Σ), denoted $mods(D, \Sigma)$, is the set of instances $\{I \mid I \supseteq D \wedge I \models \Sigma\}$. We say that D *entails* (q, Σ), denoted $D \models (q, \Sigma)$, if $I \models Q$ for every $I \in mods(D, \Sigma)$. *Ontology-mediated query answering (OMQA)* is the task of deciding whether $D \models (q, \Sigma)$ for a given database D and an OMQ (q, Σ). A different way to define the existential rules semantics is via the concept of the *Chase* (see, e.g., [20, 21]).

Following Vardi [22], the *combined complexity* of BCQ answering considers the database, the set of dependencies, and the query as part of the input. The *bounded-arity-combined* (or *ba-combined*) complexity assumes that the arity of the underlying schema is bounded by an

Table 1
Complexity of OMQA under existential rules [19].

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

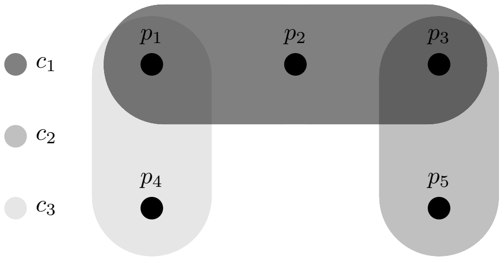

Figure 1
Protein containment in complexes.

integer constant. The *fixed-program-combined* (or *fp-combined*) *complexity* considers the sets of TGDs as fixed; the *data complexity* also assumes the query fixed. Table 1 summarizes the complexity results for OMQA in the different TGD classes here considered; OMQA ($\mathcal{L}$) denotes the OMQA problem when restricted over ontologies belonging to $\mathcal{L}$.

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

3. Explanations for Ontology-Mediated Query Answers

In our approach, a minimal explanation for an ontology-mediated query answer is an inclusion-minimal subset of a database that supports such query answer.

Definition 1. For a database D and an OMQ (q, Σ), where q is a query and Σ is a program, such that $D \models (q, \Sigma)$, an *explanation for* $D \models (q, \Sigma)$ is a subset $E \subseteq D$ of facts entailing the OMQ (q, Σ), i.e., $E \models (q, \Sigma)$; E is a *minimal explanation*, or *MinEx*, *for* $D \models (q, \Sigma)$ if E is *subset-minimal*, i.e., there is no proper subset $E' \subsetneq E$ that is an explanation for $D \models (q, \Sigma)$.

Example 2. Each protein $p_1, \ldots, p_5$ belongs to complexes c_1, c_2 and c_3, as shown in Fig 1. We want to find minimal subsets of proteins covering all complexes (a minimal hitting set [23, 24]). This is tantamount to looking for a MinEx, where the database is $D = \{prt(p_i) \mid 1 \leq i \leq 5\} \cup \{in(p_i, c_j) \mid p_i \text{ is in } c_j\}$, the ontology is $\Sigma = \{prt(P) \wedge in(P, C) \rightarrow cov(C)\}$, and the query is $q = (cov(c_1) \wedge cov(c_2) \wedge cov(c_3)) \wedge \phi_{in}$, where ϕ_{in} is a conjuction of all *in* facts in D. Clearly, the MinExes for (q, Σ) in D are in bijection with the protein covers of complexes. ◁

We analyze these problems. **Is-MinEX**: given a set of facts, decide if it is a *MinEx*; **All-MinEX**: given sets of facts, decide if they are *all the MinExes*; **MinEX-Rel**: given a fact, decide if it is *contained in some MinEx*; **MinEX-Irrel**: given forbidden sets of facts, decide if there is a *MinEx not containing any forbidden set*; and **Small-MinEX** (resp., **Large-MinEX**): given a number n, decide if there is a MinEx with *less (resp. more) elements than* n.

Example 3. Consider the following subsets of the database:

$$E_1 = \{prt(p_1), prt(p_3)\} \cup D_{in}, \text{ and } E_2 = \{prt(p_2), prt(p_4), prt(p_5)\} \cup D_{in},$$

Table 2
Complexity results for Is-MinEX ($\mathcal{L}$) and for All-MinEX ($\mathcal{L}$).

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

where D_{in} is the set of all in facts from D. Then we have:

- Is-MinEX: Both E_1 and E_2 are MinExes for (q, Σ) in D . On the other hand, $E_1' = \{prt(p_1), prt(p_4)\} \cup D_{in}$ and $E_2' = \{prt(p_1), prt(p_2), prt(p_3)\} \cup D_{in}$ are not MinExes as they do not entail the query and are not minimal, respectively.
- All-MinEX: Both $E_3 = \{prt(p_1), prt(p_5)\} \cup D_{in}$ and $E_4 = \{prt(p_3), prt(p_4)\} \cup D_{in}$ are MinExes, and the set $\mathcal{E} = \{E_1, E_2, E_3, E_4\}$ is the set of all MinExes for (q, Σ) in D .
- MinEX-Rel: For each $i \in \{1, \ldots, 5\}$, the fact $prt(p_i)$ is contained in some MinEx for (q, Σ) in D , and thus the fact is relevant.
- MinEX-Irrel: Let $\{\{prt(p_1), prt(p_3)\}, \{prt(p_5)\}\}$ be a set of forbidden sets of facts. Note that there is an explanation that does not contain any of these sets, which is E_4. Notice, however, E_1, E_2 and E_3 contain a forbidden set.
- Large-MinEX and Small-MinEX: All MinExes are either of size $2 + |D_{in}|$ or $3 + |D_{in}|$. $\lhd$

4. Recognizing (Sets of All) Minimal Explanations

First, we provide general results for Is-MinEX and All-MinEX. These general algorithms use the procedure for the OMQA problem. For Is-MinEX, checking that a set E is a MinEx requires checking that E entails the OMQ, and that E is *minimal*, if removing any element from E yields a set that does not entail the OMQ. For All-MinEX, we need to check that each set of facts in the input entails the OMQ, and that in the input there is no missing explanation.

Theorem 4. *For a class $\mathcal{L}$ of TGDs, if OMQA ($\mathcal{L}$) is in the complexity class $\mathcal{C}$, then:*

- *Is-MinEX($\mathcal{L}$) can be decided by a check in $\mathcal{C}$ and a linear number of checks in co-$\mathcal{C}$;*
- *All-MinEX($\mathcal{L}$) can be decided by a check in $\mathrm{P}^{\mathcal{C}}$ and a check in co-($\mathrm{NP}^{\mathcal{C}}$).*

Theorem 5. *If $\mathcal{L}$ is a class of TGDs for which OMQA is FO-rewritable, then Is-MinEX($\mathcal{L}$) is in AC^0 in the data complexity.*

Theorem 6. *For a class $\mathcal{L}$ of TGDs, if OMQA($\mathcal{L}$) is in NP, then All-MinEX($\mathcal{L}$) is in D^{P}.*

Theorem 7. *Let $\mathcal{L}$ be class of TGDs for which OMQA is FO-rewritable. Then, computing all MinExes for $D \models (q, \Sigma)$, where Σ is from $\mathcal{L}$, is feasible in polynomial time in the data complexity.*

Table 3
Complexity results for MinEX-Irrel/Small-MinEX ($\mathcal{L}$) and for MinEX-Rel/Large-MinEX ($\mathcal{L}$).

	MinEX-Irrel / Small-MinEX				MinEX-Rel / Large-MinEX			
$\mathcal{L}$	**Data**	fp**-comb.**	ba**-comb.**	**Comb.**	**Data**	fp**-comb.**	ba**-comb.**	**Comb.**
L, LF, AF	in P	NP	NP	PSPACE	in P	Σ_2^{P}	Σ_2^{P}	PSPACE
S, SF	in P	NP	NP	EXP	in P	Σ_2^{P}	Σ_2^{P}	EXP
A	in P	NP	NEXP	NEXP	in P	Σ_2^{P}	P^{NEXP}	P^{NEXP}
G	NP	NP	EXP	2EXP	NP	Σ_2^{P}	EXP	2EXP
F, GF	NP	NP	NP	EXP	NP	Σ_2^{P}	Σ_2^{P}	EXP
WS, WA	NP	NP	2EXP	2EXP	NP	Σ_2^{P}	2EXP	2EXP

The upper-bounds stated above can be shown tight. For Is-MinEX, the P-hardness results are obtained via a reduction from OMQA for GF, which is P-complete in those cases; the D^{P}-hardness results via a reduction from the classical Sat-UnSat problem; and the D^{EXP}-hardness results via a reduction from the D^{EXP}-complete problem ExpTiling-NonExpTiling: given two tiling problems TP_1 and TP_2 for the exponential square $2^n \times 2^n$, and two initial tiling conditions w_1 and w_2, decide whether TP_1 have no solution with w_1 *and* TP_2 have a solution with w_2.

All the remaining hardness results in Table 2 for Is-MinEX follow from the observation that Is-MinEX is no easier than OMQA in the *ba*- and combined complexity.

For All-MinEX, the co-NP-hardness results are via a reduction from UnSat, while all other hardness results are obtained with minimal variations on the Is-MinEX's hardness proofs.

5. Existence of an Explanation Excluding/Including Facts

For MinEX-Irrel, to check the existence of a MinEx not intersecting any of the forbidden sets provided in input, it suffices to guess a set of fact entailing the OMQ (we do not need to check the minimality of the guessed set in this case). For MinEX-Rel, checking the existence of a MinEx including a fact we need to guess a set of facts that subsequently needs to be checked to be a MinEx (we do need to check the minimality of the guessed set in this case).

Theorem 8. *For a class $\mathcal{L}$ of TGDs, if OMQA ($\mathcal{L}$) (resp., Is-MinEX ($\mathcal{L}$)), is in the complexity class $\mathcal{C}$ (resp., $\mathcal{D}$), then MinEX-Irrel$(\mathcal{L})$ (resp., MinEX-Rel ($\mathcal{L}$)) is in $\text{NP}^{\mathcal{C}}$ (resp., $\text{NP}^{\mathcal{D}}$).*

Theorem 9. *For a class $\mathcal{L}$ of TGDs, if OMQA ($\mathcal{L}$) is in* NP, *then MinEX-Irrel$(\mathcal{L})$ is in* NP.

Also these complexity results are tight. In particular, for MinEX-Irrel, the NP-hardness results in the data complexity are obtained via a reduction from the problem of deciding whether in a graph there is a path between two vertices avoiding a set of pairs of edges [25, Problem GT54].

For MinEX-Rel, the NP-hardness results are via a reduction from the problem of deciding whether in a directed graph there is a path from a node s to a node t passing through a node m [26]; the Σ_2^{P}-hardness results via a reduction from the validity problem $\exists X \forall Y \phi(X, Y)$; and the P^{NEXP}-hardness results via a reduction from the following problem [19]: for a triple (m, TP_1, TP_2), where m is an integer in unary notation, and TP_1 and TP_2 are two tiling problems for the exponential square $2^n \times 2^n$, decide whether there is an initial condition w of length m, such that TP_1 has no solution with w, *and* TP_2 has a solution with w.

The other hardness results for MinEX-Irrel (resp., MinEX-Rel) follow via a reduction from OMQA in the *fp*-, *ba*-, and combined complexity (resp., in the *ba*- and combined complexity).

6. Cardinality-Based Explanation Problems

The intuitions for the membership results of Small-MinEX and Large-MinEX problems are as follows. For Small-MinEX, deciding if there is a MinEx not bigger than a given size, it suffices to guess a small enough set of facts, and then test that this set entails the OMQ (in this case, we do not need to ensure the minimality of the guessed set). For Large-MinEX, to decide the existence of a big enough MinEx, we can guess a proper set of facts and check that it is actually a MinEx (in this case, we do need to check the minimality of the guessed set).

Theorem 10. *For a class $\mathcal{L}$ of TGDs, if OMQA ($\mathcal{L}$) (resp., Is-MinEX ($\mathcal{L}$)), is in the complexity class $\mathcal{C}$ (resp., $\mathcal{D}$), then Small-MinEX ($\mathcal{L}$) (resp., Large-MinEX ($\mathcal{L}$)) is in* $\text{NP}^{\mathcal{C}}$ *(resp.,* $\text{NP}^{\mathcal{D}}$*).*

Some entries in Table 3 need the result below; P membership results follow from Theorem 7.

Theorem 11. *For a class $\mathcal{L}$ of TGDs, if OMQA ($\mathcal{L}$) is in* NP*, then Small-MinEX ($\mathcal{L}$) is in* NP*.*

We have completeness results also for these problems. For Small-MinEX, the NP-hardness results in the data complexity are via a reduction from Vertex Cover.

For Large-MinEX, the NP-hardness results are obtained via a reduction from the problem Hamiltonian Path; the Σ_2^{P}-hardness results and the P^{NEXP}-hardness are obtained via small variations of the corresponding MinEX-Rel's hardness proofs.

The other hardness results for Small-MinEX (resp., Large-MinEX) are via a reduction from OMQA in the *fp*-, *ba*-, and combined complexity (resp., in the *ba*- and combined complexity).

7. Summary and Outlook

In this paper, we have studied the problem of explaining query answers in terms of minimal subsets of database facts, and provided a thorough complexity analysis for several decision problems associated with minimal explanations under existential rules.

Future work can include studying other ontology languages. Our work on explaining OMQA has been extended to the inconsistent setting [27, 28], to negative OMQA [29], to preferred explanations [30], and also to DL [31]. The explanation notion, in line with the inconsistent setting above, can also be extended to the cases of cardinality-based repairs [32], generalized repairs [33], probabilistic/preferred repairs [34, 35, 36], and repairs based on value updates [37]. Inspired by the idea of exploring preferences over explanations, we can also consider how more elaborate preference models can be included in this framework [38, 39, 40, 41].

Acknowledgments

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

References

[1] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Explanations for query answers under existential rules, in: IJCAI, 2019, pp. 1639–1646.

[2] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, in: Journal on Data Semantics X, Springer-Verlag, 2008, pp. 133–173.

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

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

[5] A. Calì, G. Gottlob, T. Lukasiewicz, A general Datalog-based framework for tractable query answering over ontologies, J. Web Sem. 14 (2012) 57–83.

[6] A. Kalyanpur, B. Parsia, M. Horridge, E. Sirin, Finding all justifications of OWL DL entailments, in: ISWC/ASWC, 2007, pp. 267–280.

[7] F. Baader, B. Suntisrivaraporn, Debugging SNOMED CT using axiom pinpointing in the description logic EL+, in: KR-MED, 2008.

[8] R. Peñaloza, B. Sertkaya, Understanding the complexity of axiom pinpointing in lightweight description logics, Artif. Intell. 250 (2017) 80–104.

[9] M. Horridge, B. Parsia, U. Sattler, Laconic and precise justifications in OWL, in: ISWC, 2008, pp. 323–338.

[10] M. Horridge, B. Parsia, U. Sattler, Explaining inconsistencies in OWL ontologies, in: SUM, 2009, pp. 124–137.

[11] R. Sebastiani, M. Vescovi, Axiom pinpointing in lightweight description logics via Horn-SAT encoding and conflict analysis, in: CADE, 2009, pp. 84–99.

[12] A. Borgida, D. Calvanese, M. Rodriguez-Muro, Explanation in the DL-Lite family of description logics, in: OTM, 2008, pp. 1440–1457.

[13] 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.

[14] D. Calvanese, M. Ortiz, M. Šimkus, G. Stefanoni, Reasoning about explanations for negative query answers in DL-Lite, J. Artif. Intell. Res. 48 (2013) 635–669.

[15] İ. İ. Ceylan, S. Borgwardt, T. Lukasiewicz, Most probable explanations for probabilistic database queries, in: IJCAI, 2017, pp. 950–956.

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

[17] A. Calì, G. Gottlob, A. Pieris, Towards more expressive ontology languages: The query answering problem, Artif. Intell. 193 (2012) 87–128.

[18] R. Fagin, P. G. Kolaitis, R. J. Miller, L. Popa, Data exchange: Semantics and query answering, Theor. Comput. Sci. 336 (2005) 89–124.

[19] 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.

[20] E. Tsamoura, D. Carral, E. Malizia, J. Urbani, Materializing knowledge bases via trigger graphs, VLDB Endow. 14 (2021) 943–956.

[21] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Exploiting equality generating dependencies in checking chase termination, VLDB Endow. 9 (2016) 396–407.

[22] M. Vardi, The complexity of relational query languages, in: STOC, 1982, pp. 137–146.

[23] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, in: LICS, 2014, pp. 43:1–43:10.

[24] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, SIAM J. Comput. 47 (2018) 456–492.

[25] M. R. Garey, D. S. Johnson, Computers and Intractability, W. H. Freeman, New York, 1979.

[26] A. S. Lapaugh, C. H. Papadimitriou, The even-path problem for graphs and digraphs, Networks 14 (1984) 507–513.

[27] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for inconsistency-tolerant query answering under existential rules, in: AAAI, 2020, pp. 2909–2916.

[28] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for negative query answers under inconsistency-tolerant semantics, in: IJCAI, 2022. To appear.

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

[30] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, C. Molinaro, A. Vaicenavičius, Preferred explanations for ontology-mediated queries under existential rules, in: AAAI, 2021, pp. 6262–6270.

[31] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Explanations for ontology-mediated query answering in description logics, in: ECAI, 2020, pp. 672–679.

[32] T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Complexity of inconsistency-tolerant query answering in Datalog+/– under cardinality-based repairs, in: AAAI, 2019, pp. 2962–2969.

[33] T. Lukasiewicz, E. Malizia, C. Molinaro, Complexity of approximate query answering under inconsistency in Datalog$^{\pm}$, in: IJCAI, 2018, pp. 1921–1927.

[34] S. Greco, C. Molinaro, Probabilistic query answering over inconsistent databases, Ann. Math. Artif. Intell. 64 (2012) 185–207.

[35] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Preference-based inconsistency-tolerant query answering under existential rules, in: KR, 2020, pp. 203–212.

[36] M. Calautti, L. Caroprese, S. Greco, C. Molinaro, I. Trubitsyna, E. Zumpano, Existential active integrity constraints, Expert Syst. Appl. 168 (2021) art. no. 114297.

[37] S. Greco, C. Molinaro, I. Trubitsyna, Computing approximate query answers over inconsistent knowledge bases, in: IJCAI, 2018, pp. 1838–1846.

[38] T. Lukasiewicz, E. Malizia, On the complexity of mCP-nets, in: AAAI, 2016, pp. 558–564.

[39] T. Lukasiewicz, E. Malizia, Complexity results for preference aggregation over (m)CP-nets: Pareto and majority voting, Artif. Intell. 272 (2019) 101–142.

[40] T. Lukasiewicz, E. Malizia, Complexity results for preference aggregation over (m)CP-nets: Max and rank voting, Artif. Intell. 303 (2022) art. no. 103636.

[41] 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.