Complexity of Inconsistency-Tolerant Query Answering in Datalog+/– under Cardinality-Based Repairs*

(Discussion Paper)

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

Querying inconsistent ontological knowledge bases is an important problem in practice, for which several inconsistency-tolerant semantics have been proposed. In these semantics, the input database is erroneous, and a repair is a maximally consistent database subset. Different notions of maximality (such as subset and cardinality maximality) have been considered. In this paper, we give a precise picture of the computational complexity of inconsistency-tolerant query answering in a wide range of Datalog+/– languages under the cardinality-based versions of three prominent repair semantics.

Keywords

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

1. Introduction

In many ontology-based applications, such as ontology-based data extraction from the Web, or ontology-based integration of different data sources, it is very likely that the data are inconsistent with the ontology, and thus inconsistency-tolerant semantics for ontology-based query answering are urgently needed. Among the most prominent ontology languages are description logics (DLs) [2] and existential rules from the context of Datalog$^{\pm}$ [3].

The most widely accepted semantics for querying inconsistent ontological knowledge bases is perhaps *consistent query answering (CQA)*, which was first developed for relational databases [4] and then generalized as the ABox repair (AR) semantics for several DLs [5]. Consistent query answering is based on the concept of repair, which is a maximal consistent subset of the input database. A fact/query is entailed by an ontological knowledge base in consistent query answering, if it is (classically) entailed by all the repairs (under the ontology). Several other repair semantics for querying inconsistent knowledge bases have recently been developed as alternatives. In the *intersection of repairs (IAR)* semantics [5], an answer is considered to be

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-19 paper [1].
thomas.lukasiewicz@cs.ox.ac.uk (T. Lukasiewicz); enrico.malizia@unibo.it (E. Malizia); andrius.vaicenavicius@cs.ox.ac.uk (A. Vaicenavičius)
0000-0002-7644-1668 (T. Lukasiewicz); 0000-0002-6780-4711 (E. Malizia)
© 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)
CEUR-WS.org/Vol-3194/paper62.pdf

valid, if it can be inferred from the intersection of the repairs (and the ontology). The *intersection of closed repairs (ICR)* [6] is another semantics, in which an answer is valid, if it can be inferred from the intersection of the closure of the repairs (and the ontology). Recently, the AR semantics was extended to the *generalized repair (GR)* semantics [7]. In the GR semantics, also ontological rules may be removed. This generalization was extended to the IAR and ICR semantics in [8]

Interestingly, the IAR and the ICR semantics can be seen as under-approximation of the AR semantics and analyzing their complexity helps to understand whether such approximations have actually lower complexities (see also [9, 10] for other approximation approaches). Beside this, a crucial advantage of the IAR and the ICR semantics is that their intersection of (closed) repairs can be materialized [11, 12], while the AR semantics exists only virtually.

The complexity of consistent query answering when the ontology is described via one of the main DLs is well-understood. Rosati [13] studied the data and combined complexity for a wide spectrum of DLs, while Bienvenu [6] identified cases for simple ontologies (within the *DL-Lite* family) for which tractable data complexity results can be obtained. In [14], the data and different types of combined complexity of consistent query answering have been studied for ontologies described via existential rules and negative constraints.

Alternative maximality notions for repairs, such as cardinality-maximal repairs [15], rather than subset-maximal ones, have been explored less. Bienvenu et al. [16] analyzed the data and the combined complexity of query answering under the AR and IAR semantics over the language *DL-Lite*$_{\mathcal{R}}$ for various notions of maximal repairs, among which maximum cardinality.

This paper continues this line of research on cardinality-maximal consistent query answering, and we analyze the complexity of the above three inconsistency-tolerant query answering semantics for a wide range of Datalog$^{\pm}$ languages and for several different complexity measures.

2. Preliminaries

We here briefly recall some basics on existential rules from the context of Datalog$^{\pm}$ [3].

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 $h'(p(\mathbf{X}, \mathbf{Z})) \in I$. A *negative constraint (NC)* ν is a first-order formula $\forall \mathbf{X}\, \varphi(\mathbf{X}) \rightarrow \bot$, where $\mathbf{X} \subseteq \mathbf{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms without nulls, and $\bot$ denotes the truth constant $false$. An instance I satisfies ν, written $I \models \nu$, if there is *no* homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$. 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. For a TGD class $\mathbb{C}$, $\mathbb{C}_\bot$ 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 $\mathcal{L}$ that we consider to guarantee decidability are among the most frequently analyzed in the literature, namely, linear (L) [3], guarded (G) [17], sticky (S) [18], and acyclic TGDs (A), along with the "weak" (proper) generalizations weakly sticky (WS) [18] and weakly acyclic TGDs (WA) [19], as well as their "full" (i.e., existential-free) proper restrictions linear full (LF), guarded full (GF), sticky full (SF), and acyclic full TGDs (AF), respectively, and full TGDs (F) in general. We also recall the following further inclusions: $\mathsf{L} \subset \mathsf{G}$ and $\mathsf{F} \subset \mathsf{WA} \subset \mathsf{WS}$. We refer to [14] 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., [20, 21]). 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 [22], the *combined complexity* of BCQ answering considers the database, the set of dependencies, and the query as part of the input. The *bounded-arity-combined* (or *ba-combined*) complexity assumes that the arity of the underlying schema is bounded by an integer constant. The *fixed-program-combined* (or *fp-combined*) *complexity* considers the sets of TGDs 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 [14].

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, an inconsistent knowledge base entails every query, 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*, which is a maximal consistent subset of the database.

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

Table 1
Complexity of BCQ answering under existential rules [14].

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

explanation" [23, 24] of the NC. By deleting from D a minimal hitting set [25, 26, 27] of facts S intersecting all culprits, we obtain a repair $R = D \setminus S$.

We now define inconsistency-tolerant semantics for a generic concept of repair maximality.

Given a knowledge base $KB = (D, \Sigma)$, a *selection* D' of KB is a database such that $D' \subseteq D$. A selection D' of KB is *consistent*, if $mods((D', \Sigma)) \neq \emptyset$. Consistent selections of knowledge bases can be ordered according to some criteria to select the more desired ones. Given a preorder $\preccurlyeq$ over a set $\mathcal{S}$ of databases, for two elements $D', D'' \in \mathcal{S}$, $D' \prec D''$ denotes that $D' \preccurlyeq D''$ and $D'' \not\preccurlyeq D'$. A database $D \in \mathcal{S}$ is $\preccurlyeq$-maximal in $\mathcal{S}$ iff there is no $D' \in \mathcal{S}$ such that $D \prec D'$.

Definition 1. A $\preccurlyeq$*-repair* of a knowledge base KB is a consistent selection of KB that is $\preccurlyeq$-maximal in the set of all the consistent selections of KB.

We now define the three different inconsistency-tolerant semantics for BCQ answering. $Rep_{\preccurlyeq}(KB)$ denotes the set of all $\preccurlyeq$-repairs of KB. 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.

Definition 2. Let KB be a knowledge base, let q be a BCQ, and let $\preccurlyeq$ be an order over the consistent selections of KB.

- KB entails q under the *ABox repair semantics and order* $\preccurlyeq$ ($\preccurlyeq$-AR), denoted by $KB \models_{\preccurlyeq\text{-}AR} q$, if, for all $D' \in Rep_{\preccurlyeq}(KB)$, $(D', \Sigma) \models q$.
- KB entails q under the *intersection of repairs semantics and order* $\preccurlyeq$ ($\preccurlyeq$-IAR), denoted by $KB \models_{\preccurlyeq\text{-}IAR} q$, if $(D^*, \Sigma) \models q$, where $D^* = \bigcap\{D' \mid D' \in Rep_{\preccurlyeq}(KB)\}$.
- KB entails q under the *intersection of closed repairs semantics and order* $\preccurlyeq$ ($\preccurlyeq$-ICR), denoted by $KB \models_{\preccurlyeq\text{-}ICR} q$, if $(D_I, \Sigma) \models q$, where $D_I = \bigcap\{Cl((D', \Sigma)) \mid D' \in Rep_{\preccurlyeq}(KB)\}$.

An interesting class of repairs are those selected by the cardinality order '$\leq$' [16]. A $\leq$-repair of a knowledge base KB is a maximum cardinality consistent selection of KB. Here, we consider only the '$\leq$' order, hence, we often call $\leq$-repairs simply repairs, and by $Rep(KB)$, we mean $Rep_{\leq}(KB)$. Cardinality-maximal repairs are very appropriate when it is known (or believed) that all the facts in the database have the same (possibly small) probability of being erroneous. In these cases, larger repairs are preferred, because fewer facts are dropped [16]. When facts in the database have different likelihoods of being erroneous, then other concepts of repairs can also be taken into consideration [6, 28].

Table 2
Complexity of $\leq$-$AR/IAR/ICR$ BCQ answering—all completeness results. $^{+}$Different membership proof for AR/IAR in [16]. *Different hardness proof for AR/IAR for $\mathsf{L}_\bot$, $\mathsf{G}_\bot$, $\mathsf{S}_\bot$, and $\mathsf{WS}_\bot$, in [16].

$\mathcal{L}$	$\leq$-AR BCQ answering: **Data**	$\leq$-AR BCQ answering: fp-**comb.**	$\leq$-AR BCQ answering: ba-**comb.**	$\leq$-AR BCQ answering: **Comb.**	$\leq$-IAR/ICR BCQ answering: **Data**	$\leq$-IAR/ICR BCQ answering: fp-**comb.**	$\leq$-IAR/ICR BCQ answering: ba-**comb.**	$\leq$-IAR/ICR BCQ answering: **Comb.**
$\mathsf{L}_\bot$, $\mathsf{LF}_\bot$, $\mathsf{AF}_\bot$	$\Theta_2^{\mathrm{P}\,+*}$	Π_2^{P}	Θ_3^{P}	PSPACE	$\Theta_2^{\mathrm{P}\,+*}$	Θ_2^{P}	Θ_3^{P}	PSPACE
$\mathsf{S}_\bot$, $\mathsf{SF}_\bot$	$\Theta_2^{\mathrm{P}\,+*}$	Π_2^{P}	Θ_3^{P}	EXP	$\Theta_2^{\mathrm{P}\,+*}$	Θ_2^{P}	Θ_3^{P}	EXP
$\mathsf{A}_\bot$	$\Theta_2^{\mathrm{P}\,+}$	Π_2^{P}	$\mathrm{P}^{\mathrm{NEXP}}$	$\mathrm{P}^{\mathrm{NEXP}}$	$\Theta_2^{\mathrm{P}\,+}$	Θ_2^{P}	$\mathrm{P}^{\mathrm{NEXP}}$	$\mathrm{P}^{\mathrm{NEXP}}$
$\mathsf{G}_\bot$	$\Theta_2^{\mathrm{P}\,+*}$	Π_2^{P}	EXP	2EXP	$\Theta_2^{\mathrm{P}\,+*}$	Θ_2^{P}	EXP	2EXP
$\mathsf{F}_\bot$, $\mathsf{GF}_\bot$	$\Theta_2^{\mathrm{P}\,+}$	Π_2^{P}	Θ_3^{P}	EXP	$\Theta_2^{\mathrm{P}\,+}$	Θ_2^{P}	Θ_3^{P}	EXP
$\mathsf{WS}_\bot$, $\mathsf{WA}_\bot$	$\Theta_2^{\mathrm{P}\,+*}$	Π_2^{P}	2EXP	2EXP	$\Theta_2^{\mathrm{P}\,+*}$	Θ_2^{P}	2EXP	2EXP
$\mathsf{WG}_\bot$	EXP	EXP	EXP	2EXP	EXP	EXP	EXP	2EXP

3. Complexity Analysis

Compared to the case where subset maximality is considered, using maximum cardinality in several cases comes at a cost, needed to compute the largest repair's size. However, this is sometimes masked out by the complexity of classical/$AR/IAR/ICR$ BCQ answering. Compared to $\leq$-AR-BCQ answering, the complexity of $\leq$-IAR- and $\leq$-ICR-BCQ answering slightly drops and is the same. Their complexity is the same because the complexity of either classical BCQ reasoning or of computing the biggest repairs' size dominate the task's complexity. Therefore, in this setting ICR has an advantage over IAR, as ICR is a finer AR's approximation than IAR.

3.1. Membership Results

A first result allows us to show most of the complexity upper-bounds of Table 2. The intuition behind this theorem is as follows. First we can compute the size of the biggest repairs via a binary search, then through some additional oracle calls it is possible to check whether the query is entailed or not under the inconsistency-tolerant semantics (see [1] for more details).

Theorem 3. *Let L be a Datalog$^\pm$ language. If BCQ answering from knowledge bases over L is in $\mathbf{C}$ in the data / ba-combined / combined complexity (resp., data / ba-combined complexity), then $\leq$-AR and $\leq$-IAR (resp., $\leq$-ICR) BCQ answering from knowledge bases over L is in* P *with an oracle for* $\mathrm{NP}^{\mathbf{C}}[O(\log n)]$ *in the data / ba-combined / combined complexity (resp., data / ba-combined complexity).*

The previous result relies on the guess of a query entailment disprover to be passed to the oracle. However, this cannot be done for the ICR case in the combined complexity, as the guess might need be too large. We hence need a tailored proof analyzing all languages case by case.

Theorem 4. *$\leq$-ICR BCQ answering in the combined complexity from knowledge bases over Datalog$^\pm$ the languages L here considered is in the complexity classes shown in Table 2.*

For the upper-bounds in the fp-combined setting, we can actually provide tighter ones, as checking the consistency of a set of facts is feasible in the complexity class of BCQ answering in the *data* complexity (and *not* in the fp-complexity), because the NCs are fixed.

Theorem 5. *If BCQ answering from knowledge bases over a Datalog$^{\pm}$ language L is in $\mathbf{D}$ in the data complexity and in $\mathbf{C}$ in the fp-combined complexity, then $\leq$-AR (resp., $\leq$-IAR and $\leq$-ICR) BCQ answering from knowledge bases over L is possible by a computation in* P *with an oracle for* $NP^{\mathbf{D}}[O(\log n)]$*, followed by a computation in* co-$NP^{\mathbf{C}}$ *(resp., $\mathbf{C}$), in the fp-combined complexity.*

3.2. Hardness Results

We can show matching lower-bounds for the upper-bounds found in the previous section.

The following result is via a reduction from the Θ_2^{p}-complete problem InAllMaxIS [15]: for a graph G and a vertex w, decide if w belongs to all the max-size independent sets of G.

Theorem 6. *For every $C \in \{AR, IAR, ICR\}$, $\leq$-C BCQ answering from knowledge bases over* $\mathsf{LF}_\perp$, $\mathsf{AF}_\perp$, *and* $\mathsf{SF}_\perp$ *is* Θ_2^{p}-hard *in the data complexity.*

For the next result, the reduction is from the classical Π_2^{p}-complete problem of deciding the validity of a QBF $\forall X \exists Y \phi(X, Y)$.

Theorem 7. *$\leq$-AR BCQ answering from knowledge bases over* $\mathsf{LF}_\perp$, $\mathsf{AF}_\perp$, *and* $\mathsf{SF}_\perp$ *is* Π_2^{p}-hard *in the fp-combined complexity.*

The Θ_3^{p}-hardness of the following problems is via a reduction from the Θ_3^{p}-complete problem Comp-Valid$_2$: given sets A and B of QBFs with 2 alternating quantifiers, decide whether the number of valid formulas in A is bigger than the number of valid formulas in B [29] (this is a generalization of the Comp-SAT problem [30]).

Theorem 8. *For every $C \in \{AR, IAR, ICR\}$, $\leq$-C BCQ answering from knowledge bases over* $\mathsf{LF}_\perp$, $\mathsf{AF}_\perp$, *and* $\mathsf{SF}_\perp$ *is* Θ_3^{p}-hard *in the ba-combined complexity.*

The next hardness is obtained via a reduction from the following $\mathrm{P}^{\mathrm{NEXP}}$-hard problem [14]: 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.

Theorem 9. *For any $C \in \{AR, IAR, ICR\}$, $\leq$-C BCQ answering for* $\mathsf{A}_\perp$ *are* P^{NEXP}*-hard in the ba-combined and combined complexity.*

The remaining hardness results follows from the hardness of BCQ answering.

4. Summary and Outlook

We have analyzed BCQ answering under different cardinality-maximal inconsistency-tolerant semantics, for the most popular Datalog$^{\pm}$ languages and complexity measures.

Future research include defining other semantics for inconsistency-tolerant ontological query answering, considering weighed repairs and more elaborate user preferences over repairs [31, 32, 33, 34, 35]. Also, in the line of a more recent research, it would be interesting to extend the concepts of explanations for inconsistency-tolerant query answering [36, 37] to cardinality-maximal repairs, and mix this with the notions of preferred explanations [38] and explanations for negative query answers [39].

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

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

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

[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] T. Eiter, T. Lukasiewicz, L. Predoiu, Generalized consistent query answering under existential rules, in: KR, 2016, pp. 359–368.

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

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

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

[11] 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.

[12] M. Bienvenu, C. Bourgaux, Inconsistency-tolerant querying of description logic knowledge bases, in: Reasoning Web, 2016, pp. 156–202.

[13] R. Rosati, On the complexity of dealing with inconsistency in description logic ontologies, in: IJCAI, 2011, pp. 1057–1062.

[14] 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.

[15] A. Lopatenko, L. E. Bertossi, Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics, in: ICDT, 2007, pp. 179–193.

[16] M. Bienvenu, C. Bourgaux, F. Goasdoué, Querying inconsistent description logic knowledge bases under preferred repair semantics, in: AAAI, 2014, pp. 996–1002.

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

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

[19] R. Fagin, P. G. Kolaitis, R. J. Miller, L. Popa, Data exchange: Semantics and query answering, Theor. Comput. Sci. 336 (2005) 89–124.
[20] 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] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Explanations for query answers under existential rules, in: IJCAI, 2019, pp. 1639–1646.
[24] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Explanations for ontology-mediated query answering in description logics, in: ECAI, 2020, pp. 672–679.
[25] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, SIAM J. Comput. 47 (2018) 456–492.
[26] J. Chomicki, J. Marcinkowski, Minimal-change integrity maintenance using tuple deletions, Inf. Comput. 197 (2005) 90–121.
[27] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, in: LICS, 2014, pp. 43:1–43:10.
[28] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Preference-based inconsistency-tolerant query answering under existential rules, in: KR, 2020, pp. 203–212.
[29] 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.
[30] T. Lukasiewicz, E. Malizia, On the complexity of mCP-nets, in: AAAI, 2016, pp. 558–564.
[31] 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.
[32] T. Lukasiewicz, E. Malizia, Complexity results for preference aggregation over (m)CP-nets: Pareto and majority voting, Artif. Intell. 272 (2019) 101–142.
[33] T. Lukasiewicz, E. Malizia, Complexity results for preference aggregation over (m)CP-nets: Max and rank voting, Artif. Intell. 303 (2022) art. no. 103636.
[34] M. Calautti, L. Caroprese, S. Greco, C. Molinaro, I. Trubitsyna, E. Zumpano, Existential active integrity constraints, Expert Syst. Appl. 168 (2021) art. no. 114297.
[35] 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.
[36] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for inconsistency-tolerant query answering under existential rules, in: AAAI, 2020, pp. 2909–2916.
[37] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for negative query answers under inconsistency-tolerant semantics, in: IJCAI, 2022.
[38] 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.
[39] 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.