On the Complexity of Querying Inconsistent Weighted Knowledge Bases

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 are approaches to provide meaningful answers to queries even in the presence of inconsistent knowledge. Several such semantics rely on the notion of a *repair*, which is a "maximal" consistent subset of the database, where different maximality criteria might be adopted depending on the application at hand. Common maximality criteria assume that all facts in a database are equally important. However, in several real-world applications, it is often the case that different facts have different importance. In this paper, we consider Datalog$^\pm$ knowledge bases where database facts are weighted, with weights expressing facts' importance (or reliability or some other aspect of interest). We present recent results on the complexity of querying inconsistent knowledge bases in this setting.

Keywords

Inconsistency-tolerant semantics, computational complexity

1. Introduction

In real-world applications, data possibly coming from different sources may exhibit inconsistencies. Obtaining meaningful query answers in these scenarios can be achieved by resorting to inconsistency-tolerant semantics. Popular ones are the *ABox repair (AR)*, first defined for relational databases [1] and then generalized for description logics (DLs) [2], the *intersection of repairs (IAR)* [2], and the *intersection of closed repairs (ICR)* [3]. All such semantics, as well as others (see, e.g., [2]), are based on the notion of a *repair*, which is a "maximal" consistent subset of the knowledge base's facts. Subset maximality was adopted upon introduction of the above semantics. However, other maximality criteria are relevant in practice and have been introduced over the years. For instance, maximum cardinality is a stronger criterion ruling out subset-maximal repairs not containing the highest number of facts, which is suitable for settings where all database facts are considered equally reliable. For Datalog$^\pm$ languages, subset-maximal repairs have been considered in [4, 5] while cardinality-maximal ones in [6]; in the context of querying inconsistent DL knowledge bases, the aforementioned maximality criteria, as well as others, have been investigated in [7]. Inconsistency-tolerant semantics have been defined also w.r.t. "preferred" repairs that are selected among the subset-maximal ones on the basis of user preferences [8, 9, 10, 11, 12].

In this paper, we consider the case where database facts are associated with weights (e.g., quantitatively measuring their reliability), a scenario arising in many applications. For example, consider a neuro-symbolic system in which the neuronal part of the system produces some predictions as database facts associated with a confidence score (see, e.g., [13] and references therein). Then, in case of inconsistencies, these values can be used in the computation of most reliable repairs. In such a setting, a natural criterion to define repairs is to select the weight-maximal consistent subsets of the database. In this paper, we

AIxIA 2024 Discussion Papers - 23rd International Conference of the Italian Association for Artificial Intelligence, Bolzano, Italy, November 25–28, 2024

*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)

discuss recent results presented in [14] on the complexity of the AR, IAR, and ICR semantics when such a notion of repair is adopted in the presence of weighted knowledge bases expressed via Datalog$^{\pm}$ languages.

2. Preliminaries

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, a null, or a 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 containing only constants and nulls. A *database* D is a finite instance that contains only constants. A *homomorphism* is a substitution $h\colon \mathbf{C} \cup \mathbf{N} \cup \mathbf{V} \to \mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$ that is the identity on $\mathbf{C}$ and maps $\mathbf{N}$ to $\mathbf{C} \cup \mathbf{N}$. A *Boolean conjunctive query* (BCQ) q has the form $\exists \mathbf{Y} \phi(\mathbf{Y})$, where $\phi(\mathbf{Y})$ is a conjunction of atoms without nulls. A BCQ q is *true* over an instance I, denoted $I \models q$, if there is a homomorphism h with $h(\phi(\mathbf{Y})) \subseteq I$.

Dependencies. A *tuple-generating dependency* (TGD) σ is a first-order formula $\forall \mathbf{X} \forall \mathbf{Y}\, (\varphi(\mathbf{X}, \mathbf{Y}) \to \exists \mathbf{Z}\, p(\mathbf{X}, \mathbf{Z}))$, where $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are pairwise disjoint sets of variables, $\varphi(\mathbf{X}, \mathbf{Y})$ is a conjunction of atoms, and $p(\mathbf{X}, \mathbf{Z})$ is an atom, all without nulls. An instance I satisfies a TGD σ, written $I \models \sigma$, if the following holds: whenever there exists a homomorphism h such that $h(\varphi(\mathbf{X}, \mathbf{Y})) \subseteq I$, then there exists $h' \supseteq h|_{\mathbf{X}}$, where $h|_{\mathbf{X}}$ is the restriction of h on $\mathbf{X}$, such that $h'(p(\mathbf{X}, \mathbf{Z})) \in I$. A *negative constraint (NC)* ν is a first-order formula $\forall \mathbf{X}\, (\varphi(\mathbf{X}) \to \bot)$, where $\mathbf{X} \subseteq \mathbf{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms without nulls, and $\bot$ denotes the truth constant false. An instance I satisfies an NC ν, written $I \models \nu$, if there is *no* homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$. We will use q_ν to denote the BCQ $\exists \mathbf{X}\, \varphi(\mathbf{X})$. 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 a class $\mathbb{C}$ of TGDs, $\mathbb{C}_\bot$ denotes the combination of $\mathbb{C}$ with arbitrary NCs. Finite sets of TGDs and NCs are called *programs*. The Datalog$^{\pm}$ languages we consider are among the most frequently analyzed in the literature, namely, linear (L) [15], guarded (G) [16], sticky (S) [17], and acyclic TGDs (A), the "weak" generalizations weakly sticky (WS) [17] and weakly acyclic TGDs (WA) [18], their "full" restrictions linear full (LF), guarded full (GF), sticky full (SF), and acyclic full TGDs (AF), respectively, and full TGDs (F) in general. We refer to [12, 5] for a detailed overview.

Knowledge Bases. A *knowledge base* is a pair (D, Σ), where D is a database and Σ is a program. For a program Σ, Σ_T and Σ_{NC} denote the subsets of Σ containing the TGDs and NCs of Σ, respectively. The set of *models* of $KB = (D, \Sigma)$, denoted $mods(KB)$, is the set of instances $\{I \mid I \supseteq D \wedge I \models \Sigma\}$. We say that KB is *consistent* if $mods(KB) \neq \emptyset$, otherwise KB is *inconsistent*. The answer to a BCQ q relative to KB is *true*, denoted $KB \models q$, if $I \models q$ for every $I \in mods(KB)$. Another way to define ontological query answering is via the concept of the *Chase* (see, e.g., [16, 19]).

The BCQ answering problem is: given a knowledge base KB and a BCQ q, decide whether $KB \models q$. Following [20], the *combined complexity* of BCQ answering considers the database, the program, 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 constant. The *fixed-program-combined* (or *fp-combined*) *complexity* considers the program fixed; in the *data complexity* the query is fixed as well. We refer to [5] for an overview of the complexity of BCQ answering for the languages in this paper. For more on computational complexity theory we refer the reader to any textbook on the topic, such as [21].

3. Inconsistency-Tolerant Semantics for Weighted KBs

From now on, we implicitly assume that the database D of any knowledge base comes along with a weight function $w\colon D \to \mathbb{N}$ assigning weights to its facts. For every $D' \subseteq D$, w assigns a weight to D' defined as $w(D') = \sum_{f \in D'} w(f)$ (with a slight abuse of notation, w applies to both facts and sets of facts). For every $D_1, D_2 \subseteq D$, we write $D_1 \leq_w D_2$ (resp., $D_1 <_w D_2$) iff $w(D_1) \leq w(D_2)$ (resp., $w(D_1) < w(D_2)$).

Given a knowledge base $KB = (D, \Sigma)$, a *selection* of KB is a database D' such that $D' \subseteq D$. A selection D' of KB is *consistent* iff (D', Σ) is consistent. Symmetrically, the concept of consistent selection is linked to that of *culprit*, which is a subset C of D s.t. $(C, \Sigma_T) \models q_\nu$ for some $\nu \in \Sigma_{NC}$. By deleting from D a hitting set ([22, 23, 24]) of facts S intersecting all culprits, we obtain a consistent selection $D' = D \setminus S$.

Definition 3.1. *A* $\leq_w$-repair *of a knowledge base KB is a consistent selection D' of KB such that there is no consistent selection D'' of KB with $D' <_w D''$.*

For a knowledge base $KB = (D, \Sigma)$, $Rep_{\leq_w}(KB)$ denotes the set of all $\leq_w$-repairs of KB, and the *closure* of KB, denoted $Cl(KB)$, is the set of all facts built from constants in D and Σ, entailed by D and the TGDs of Σ.

Definition 3.2. *Let KB be a knowledge base and let q be a BCQ.*

- *KB entails q under the $\leq_w$-AR semantics, denoted $KB \models_{\leq_w\text{-}AR} q$, if $(D', \Sigma) \models q$ for all $D' \in Rep_{\leq_w}(KB)$.*
- *KB entails q under the $\leq_w$-IAR semantics, denoted $KB \models_{\leq_w\text{-}IAR} q$, if $(D_I, \Sigma) \models q$, where $D_I = \bigcap\{D' \mid D' \in Rep_{\leq_w}(KB)\}$.*
- *KB entails q under the $\leq_w$-ICR semantics, denoted $KB \models_{\leq_w\text{-}ICR} q$, if $(D_C, \Sigma) \models q$, where $D_C = \bigcap\{Cl((D', \Sigma)) \mid D' \in Rep_{\leq_w}(KB)\}$.*

4. Discussion of Complexity Results

The problems whose complexity we are interested in are denoted as $\leq_w$-$S(\mathcal{L})$, with $S \in \{AR, IAR, ICR\}$, and are defined as follows: Given a knowledge base (D, Σ) with $\Sigma \in \mathcal{L}$, and a BCQ q, does $(D, \Sigma) \models_{\leq_w\text{-}S} q$ hold?

The complexity results are summarized in Tables 1 and 2. All entries are completeness results. The complexity ranges from Δ_2^{P}- to 2EXP-completeness. For more details on how the results have been derived, we refer the reader to [14]. Here we focus on the main takeaways the complexity analysis provides.

The IAR and ICR semantics have the same complexity, which is a behavior shown by cardinality-maximal repairs as well [6], while this does not hold for subset-maximal ones [5]. As usual (under other maximality criteria to define repairs), the IAR and ICR semantics are at most as expensive as the AR semantics. Indeed, we can see that the complexity increases when moving from the IAR/ICR to the AR semantics only in the fixed-program combined complexity, while the complexity does not change across the three inconsistency-tolerant semantics under the remaining complexity measures (namely, data, bounded-arity combined, and combined complexity).

It is also interesting to compare weight-maximal repairs with subset-maximal and cardinality-maximal ones, whose complexity results can be found in [5] and [6], respectively. Clearly, weight-maximal repairs generalize cardinality-maximal ones (the latter can be simply modeled by assigning the same weight to all facts), and when we move from the latter to the former, the complexity of all inconsistency-tolerant semantics increases in several cases. Compared with subset-maximal repairs, the complexity of all inconsistency-tolerant semantics under weight-maximal repairs is always at least as high as the one under subset-maximal repairs. Overall, we can conclude that while weights give us the flexibility of assigning different importance to different facts, they incur an increase of complexity compared with more "standard" notions of repairs.

$\mathcal{L}$	**Data**	fp-**c.**	ba-**c.**	**Comb.**
$\mathsf{L}_\perp, \mathsf{LF}_\perp, \mathsf{AF}_\perp$	Δ_2^{P}	Π_2^{P}	Δ_3^{P}	PSPACE
$\mathsf{S}_\perp, \mathsf{SF}_\perp$	Δ_2^{P}	Π_2^{P}	Δ_3^{P}	EXP
$\mathsf{A}_\perp$	Δ_2^{P}	Π_2^{P}	$\mathrm{P}^{\mathrm{NEXP}}$	$\mathrm{P}^{\mathrm{NEXP}}$
$\mathsf{G}_\perp$	Δ_2^{P}	Π_2^{P}	EXP	2EXP
$\mathsf{F}_\perp, \mathsf{GF}_\perp$	Δ_2^{P}	Π_2^{P}	Δ_3^{P}	EXP
$\mathsf{WS}_\perp, \mathsf{WA}_\perp$	Δ_2^{P}	Π_2^{P}	2EXP	2EXP

Table 1
Complexity of $\leq_w$-$AR(\mathcal{L})$.

$\mathcal{L}$	**Data**	fp-**c.**	ba-**c.**	**Comb.**
$\mathsf{L}_\perp, \mathsf{LF}_\perp, \mathsf{AF}_\perp$	Δ_2^{P}	Δ_2^{P}	Δ_3^{P}	PSPACE
$\mathsf{S}_\perp, \mathsf{SF}_\perp$	Δ_2^{P}	Δ_2^{P}	Δ_3^{P}	EXP
$\mathsf{A}_\perp$	Δ_2^{P}	Δ_2^{P}	$\mathrm{P}^{\mathrm{NEXP}}$	$\mathrm{P}^{\mathrm{NEXP}}$
$\mathsf{G}_\perp$	Δ_2^{P}	Δ_2^{P}	EXP	2EXP
$\mathsf{F}_\perp, \mathsf{GF}_\perp$	Δ_2^{P}	Δ_2^{P}	Δ_3^{P}	EXP
$\mathsf{WS}_\perp, \mathsf{WA}_\perp$	Δ_2^{P}	Δ_2^{P}	2EXP	2EXP

Table 2
Complexity of $\leq_w$-$IAR(\mathcal{L})$ and $\leq_w$-$ICR(\mathcal{L})$.

5. Summary and Outlook

We have considered the problem of querying inconsistent knowledge bases whose database facts are weighted. We have discussed recent results, presented in [14], on the complexity of inconsistent-tolerant semantics in such a setting.

Future research includes defining other semantics for inconsistency-tolerant OMQA, by considering more elaborate user preferences over repairs [25, 26, 27, 28, 29, 30, 31] and also considering compact representations [32, 33, 34]. Another interesting approach that has been investigated recently in the context of handling inconsistent knowledge is that of measuring inconsistencies via the Shapley value [35], it would be interesting to bring to existential rules the ideas implemented for DLs [36, 37, 38]. As a natural extension of the setting considered in this paper, TGDs and NCs might be weighted too, similarly to what has been recently done in [39], which considers weighted knowledge bases where both axioms and assertions have weights. Another direction for future work is to devise approximation algorithms that are practical, as done in the setting of *incomplete* databases [40, 41], e.g. by resorting to a logic programming approach [42]. Recently, there has been an increasing interest on explainable AI, including explaining query answering under existential rules [43, 44, 45] and DLs [46, 47]. In particular, [46, 48, 49] addressed the problem of explaining why a query is entailed or not under inconsistency-tolerant semantics, where repairs are subset-maximal. An interesting direction for future work is to address the same problem for weight-maximal repairs.

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 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.

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] T. Lukasiewicz, E. Malizia, C. Molinaro, Complexity of approximate query answering under inconsistency in Datalog+/–, in: Proc. IJCAI, 2018, pp. 1921–1927.

[5] 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.
[6] 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.
[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] S. Staworko, J. Chomicki, J. Marcinkowski, Prioritized repairing and consistent query answering in relational databases, Ann. Math. Artif. Intell. 64 (2012) 209–246.
[9] M. Bienvenu, C. Bourgaux, Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation, in: Proc. KR, 2020, pp. 141–151.
[10] M. Bienvenu, C. Bourgaux, Querying inconsistent prioritized data with ORBITS: Algorithms, implementation, and experiments, in: Proc. KR, 2022.
[11] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Preference-based inconsistency-tolerant query answering under existential rules, in: Proc. KR 2020, 2020, pp. 203–212.
[12] M. Calautti, S. Greco, C. Molinaro, I. Trubitsyna, Preference-based inconsistency-tolerant query answering under existential rules, Artif. Intell. 312 (2022) 103772.
[13] E. Tsamoura, T. M. Hospedales, L. Michael, Neural-symbolic integration: A compositional perspective, in: Proc. AAAI, 2021, pp. 5051–5060.
[14] T. Lukasiewicz, E. Malizia, C. Molinaro, Complexity of inconsistency-tolerant query answering in datalog+/- under preferred repairs, in: Proc. KR, 2023, pp. 472–481.
[15] A. Calì, G. Gottlob, T. Lukasiewicz, A general Datalog-based framework for tractable query answering over ontologies, J. Web Sem. 14 (2012) 57–83.
[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] E. Tsamoura, D. Carral, E. Malizia, J. Urbani, Materializing knowledge bases via trigger graphs, Proc. VLDB Endow. 14 (2021) 943–956.
[20] M. Y. Vardi, The complexity of relational query languages, in: Proc. STOC, 1982, pp. 137–146.
[21] S. Arora, B. Barak, Computational Complexity: A Modern Approach, Cambridge University Press, Cambridge, UK, 2009.
[22] J. Chomicki, J. Marcinkowski, Minimal-change integrity maintenance using tuple deletions, Inf. Comput. 197 (2005) 90–121.
[23] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, in: Proc. CSL-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] T. Lukasiewicz, E. Malizia, On the complexity of mCP-nets, in: Proc. AAAI, 2016, pp. 558–564.
[26] T. Lukasiewicz, E. Malizia, Complexity results for preference aggregation over (m)CP-nets: Pareto and majority voting, Artif. Intell. 272 (2019) 101–142.
[27] T. Lukasiewicz, E. Malizia, Complexity results for preference aggregation over (m)CP-nets: Max and rank voting, Artif. Intell. 303 (2022) art. no. 103636.
[28] M. Calautti, L. Caroprese, S. Greco, C. Molinaro, I. Trubitsyna, E. Zumpano, Existential active integrity constraints, Expert Syst. Appl. 168 (2021) 114297.
[29] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On preferences and priority rules in abstract argumentation, in: Proc. IJCAI, 2022, pp. 2517–2524.
[30] 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.
[31] 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.

[32] E. Malizia, L. Palopoli, F. Scarcello, Infeasibility certificates and the complexity of the core in coalitional games, in: Proc. IJCAI, 2007, pp. 1402–1407.
[33] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, On the complexity of compact coalitional games, in: Proc. IJCAI, 2009, pp. 147–152.
[34] 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.
[35] L. S. Shapley, A value for n-person games, in: H. W. Kuhn, A. W. Tucker (Eds.), Contributions to the Theory of Games, Volume II, Princeton University Press, Princeton, NJ, USA, 1953, pp. 307–317.
[36] X. Deng, V. Haarslev, N. Shiri, Measuring inconsistencies in ontologies, in: The Semantic Web: Research and Applications, 2007, pp. 326–340.
[37] M. Bienvenu, D. Figueira, P. Lafourcade, When is shapley value computation a matter of counting?, Proc. ACM Manag. Data 2 (2024) 105.
[38] M. Bienvenu, D. Figueira, P. Lafourcade, Shapley Value Computation in Ontology-Mediated Query Answering, Technical Report arXiv:2407.20058, CoRR, 2024.
[39] M. Bienvenu, C. Bourgaux, R. Jean, Cost-based semantics for querying inconsistent weighted knowledge bases, in: Proc. KR, 2024.
[40] M. Console, P. Guagliardo, L. Libkin, Approximations and refinements of certain answers via many-valued logics, in: Proc. KR, 2016, pp. 349–358.
[41] S. Greco, C. Molinaro, I. Trubitsyna, Approximation algorithms for querying incomplete databases, Inf. Syst. 86 (2019) 28–45.
[42] S. Greco, C. Molinaro, I. Trubitsyna, E. Zumpano, NP datalog: A logic language for expressing search and optimization problems, Theory Pract. Log. Program. 10 (2010) 125–166.
[43] İ. İ. Ceylan, T. Lukasiewicz, E. Malizia, A. Vaicenavičius, Explanations for query answers under existential rules, in: Proc. IJCAI, 2019, pp. 1639–1646.
[44] 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.
[45] 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.
[46] 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.
[47] 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.
[48] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for inconsistency-tolerant query answering under existential rules, in: Proc. AAAI, 2020, pp. 2909–2916.
[49] T. Lukasiewicz, E. Malizia, C. Molinaro, Explanations for negative query answers under inconsistency-tolerant semantics, in: Proc. IJCAI, 2022, pp. 2705–2711.