From Classical to Consistent Query Answering under Existential Rules

Thomas Lukasiewicz[1], Maria Vanina Martinez[2], Andreas Pieris[3], and Gerardo I. Simari[2]

[1] Department of Computer Science, University of Oxford, UK
thomas.lukasiewicz@cs.ox.ac.uk
[2] Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur and CONICET, Argentina {mvm,gis}@cs.uns.edu.ar
[3] Institute of Information Systems, Vienna University of Technology, Austria
pieris@dbai.tuwien.ac.at

Abstract. Querying inconsistent ontologies is an intriguing new problem that gave rise to a flourishing research activity in the description logic (DL) community. The computational complexity of consistent query answering under the main DLs is rather well understood; however, little is known about existential rules. The goal of the current work is to perform an in-depth analysis of the complexity of consistent query answering under the main decidable classes of existential rules enriched with negative constraints. Our investigation focuses on the standard inconsistency-tolerant semantics, namely, the AR semantics. We establish generic complexity results, which demonstrate the tight connection between classical and consistent query answering. These results allow us to obtain in a uniform way a relatively complete picture of the complexity of our problem.

1 Introduction

An ontology is an explicit specification of a conceptualization of an area of interest. One of the main applications of ontologies is in ontology-based data access (OBDA), where they are used to enrich the extensional data with intensional knowledge. In this setting, description logics (DLs) and rule-based formalisms such as existential rules are popular ontology languages, while conjunctive queries (CQs) form the central querying tool. In real-life applications, involving large amounts of data, it is possible that the data are inconsistent with the ontology. Since standard ontology languages adhere to the classical FOL semantics, inconsistencies are nothing else than logical contradictions. Thus, the classical inference semantics fails terribly when faced with an inconsistency, since everything follows from a contradiction. This demonstrates the need for developing inconsistency-tolerant semantics.

There has been a recent and increasing focus on the development of such semantics for query answering purposes. Consistent query answering, first developed for relational databases [1] and then generalized as the AR semantics for several DLs [9], is the most widely accepted semantics for querying inconsistent ontologies. The AR semantics is

based on the idea that an answer is considered to be valid if it can be inferred from each of the repairs of the extensional data set D, i.e., the $\subseteq$-maximal consistent subsets of D. The complexity of query answering under the AR semantics when the ontology is described using one of the central DLs is rather well understood. The data and combined complexity were studied in [11] for a wide spectrum of DLs, while the work [2] identifies cases for simple ontologies (within the *DL-Lite* family) for which tractable data complexity results can be obtained. On the other hand, little is known when the ontology is described using existential rules (a.k.a. tuple-generating dependencies (TGDs) and Datalog$^{\pm}$ rules), that is, formulas of the form $\forall \mathbf{X} \forall \mathbf{Y}(\varphi(\mathbf{X}, \mathbf{Y}) \rightarrow \exists \mathbf{Z}(\psi(\mathbf{X}, \mathbf{Z})))$, and negative constraints (NCs) of the form $\forall \mathbf{X}(\varphi(\mathbf{X}) \rightarrow \bot)$, where $\bot$ denotes the truth constant *false*.

Our main goal in this work is to perform an in-depth analysis of the data and combined complexity of consistent query answering under the main decidable classes of existential rules, enriched with negative constraints. Let us recall that the main (syntactic) conditions on existential rules that guarantee the decidability of query answering are guardedness [3], stickiness [4] and acyclicity. Interestingly, our complexity analysis shows that a systematic and uniform way for transferring complexity results from classical to consistent query answering can be formally established.

To briefly summarize the main contributions:

- We present generic complexity results, which demonstrate the tight connection between classical and consistent query answering (Theorems 1 and 2).
- By exploiting our generic theorems, we obtain a (nearly) complete picture of the combined and data complexity of consistent query answering (Table 2).

For more details we refer the reader to [10].

2 Consistent Query Answering

In the classical setting of CQ answering, given a database D and a set Σ of TGDs and NCs, if the models of D and Σ, denoted $mods(D, \Sigma)$, is empty, then every query is entailed since everything is inferred from a contradiction.

Example 1. Consider the database $D = \{professor(\text{John}), fellow(\text{John})\}$, asserting that John is both a professor and a fellow, and the set Σ of TGDs and NCs consisting of

$$\begin{aligned} \forall X(professor(X) &\rightarrow \exists Y(faculty(X) \wedge teaches(X, Y))) \\ \forall X(fellow(X) &\rightarrow faculty(X)) \\ \forall X(professor(X) \wedge fellow(X) &\rightarrow \bot), \end{aligned}$$

expressing that each professor is a faculty member who teaches a course, each fellow is a faculty member, and professors and fellows form disjoint sets. It is easy to see that $mods(D, \Sigma) = \varnothing$, since John violates the disjointness constraint; thus, for every (Boolean) CQ q, $(D \wedge \Sigma) \models q$. ∎

As said above, the AR semantics is the standard semantics for querying inconsistent ontologies. A key notion, which is necessary for defining the AR semantics, is that of repair, which is a $\subseteq$-maximal consistent subset of the given database.

Definition 1. Consider a database D, and a set Σ of TGDs and NCs. A *repair* of D and Σ is some $D' \subseteq D$ such that (i) $mods(D', \Sigma) \neq \varnothing$; and (ii) there is no $\underline{a} \in (D \setminus D')$ for which $mods(D' \cup \{\underline{a}\}, \Sigma) \neq \varnothing$. Let $rep(D, \Sigma)$ be the set of repairs of D and Σ.

Example 2. Consider the database D and the set Σ of TGDs and NCs given in Example 1. The set of repairs of D and Σ consists of the following subsets of D:

$$D_1 = \{professor(\text{John})\} \qquad D_2 = \{fellow(\text{John})\}.$$

Clearly, we simply need to remove one of the database atoms in order to satisfy the single negative constraint occurring in Σ. ∎

The AR semantics [9] is based on the idea that a query can be considered to hold if it can be inferred from each of the repairs.

Definition 2. Consider a database D, a set Σ of TGDs and NCs, and a Boolean CQ q. We say that q is entailed by D and Σ under the *AR semantics*, written $(D \wedge \Sigma) \models_{AR} q$, if $(D' \wedge \Sigma) \models q$, for every $D' \in rep(D, \Sigma)$.

Example 3. Consider the database D and the set Σ of TGDs and NCs given in Example 1, and also the Boolean CQs

$$q_1 = faculty(\text{John}) \qquad q_2 = \exists X(teaches(\text{John}, X)),$$

where q_1 asks whether John is a faculty member, while q_2 asks whether John teaches a course. Recall that $rep(D, \Sigma)$ consists of the databases D_1 and D_2 given in Example 2. Clearly, $(D_i \wedge \Sigma) \models q_1$, for each $i \in \{1, 2\}$, and thus $(D \wedge \Sigma) \models_{AR} q_1$. However, even if $(D_1 \wedge \Sigma) \models q_2$, $(D_2 \wedge \Sigma) \not\models q_2$, and therefore $(D \wedge \Sigma) \not\models_{AR} q_2$. ∎

In the sequel, we refer to the problem of consistent (Boolean) CQ answering under the AR semantics as AR-CQ answering.

3 Generic Complexity Results

We present two generic complexity results that demonstrate the tight connection between classical and consistent CQ answering. These results will automatically provide us with a (nearly) complete picture of the combined and data complexity of AR-CQ answering under the main classes of TGDs, enriched with NCs. Given a class $\mathbb{C}$ of TGDs, let $\mathbb{C}_\perp$ be the formalism obtained by combining $\mathbb{C}$ with arbitrary negative constraints.

3.1 Combined Complexity

We first focus on the combined complexity. Since we would like to understand how the complexity of our problem is affected when some key parameters are fixed, we also consider the following two variants of the combined complexity: (1) the bounded-arity combined complexity (ba-combined complexity), which is calculated by assuming that the arity of the underlying schema is bounded; and (2) the fixed-program combined complexity (fp-combined complexity), which is calculated by considering the set of TGDs and negative constraints as fixed. We show the following:

Theorem 1. *Assume that CQ answering under a class $\mathbb{C}$ of TGDs is $\mathcal{C}$-complete in (x-)combined complexity, where $x \in \{\mathsf{ba}, \mathsf{fp}\}$. Then, the ($x$-)combined complexity of AR-CQ answering under $\mathbb{C}_\bot$ is (1) Π_2^p-complete, if $\mathcal{C} = \text{NP}$; and (2) $\mathcal{C}$-complete, if $\mathcal{C} \supseteq \text{PSPACE}$ is a deterministic class.*

Proof (sketch). Fix a database D, a set $\Sigma \in \mathbb{C}_\bot$ of TGDs and NCs, and a CQ q. The problem of deciding whether $(D \wedge \Sigma) \not\models_{AR} q$ can be easily solved via a guess-and-check algorithm. We simply need to apply the following steps:

1. Guess an instance $D' \subseteq D$;
2. Verify that $D' \in \mathit{rep}(D, \Sigma)$; and
3. Verify that $(D' \wedge \Sigma) \not\models q$.

We can show that steps 2 and 3 are not harder than classical query answering, which implies that AR-CQ answering under $\mathbb{C}_\bot$ is in $\text{coNP}^{\mathcal{C}}$. Therefore, (1) If $\mathcal{C} = \text{NP}$, then we get a Π_2^p upper bound since $\text{NP}^{\text{NP}} = \Sigma_2^p$ and $\text{co}\Sigma_2^p = \Pi_2^p$; and (2) If $\mathcal{C} \supseteq \text{PSPACE}$ is a deterministic class, then we get a $\mathcal{C}$ upper bound since $\text{NP}^{\mathcal{C}} = \mathcal{C}$ and $\text{co}\mathcal{C} = \mathcal{C}$.

Regarding the lower bounds, the $\mathcal{C}$-hardness result, when $\mathcal{C}$ is deterministic class above PSPACE, follows immediately since CQ answering is a special case of AR-CQ answering. For the Π_2^p-hardness, we show, by a reduction from the validity problem of 2QBF formulas, that AR-CQ answering under a single negative constraint $\forall \mathbf{X}(\varphi(\mathbf{X}) \rightarrow \bot)$, where φ consists of two atoms and it uses a single ternary predicate, while the database and the query use only binary and ternary predicates, is already Π_2^p-hard. □

3.2 Data Complexity

By providing a similar analysis as above, we can establish the following generic data complexity result:

Theorem 2. *Assume that CQ answering under a class $\mathbb{C}$ of TGDs is $\mathcal{C}$-complete in data complexity. Then, the data complexity of AR-CQ answering under $\mathbb{C}_\bot$ is (1)* coNP*-complete, if $\mathcal{C} \subseteq$ PTIME; and (2) $\mathcal{C}$-complete, if $\mathcal{C} \supseteq$ PSPACE is a deterministic class.*

Let us say that AR-CQ answering under a single negative constraint of the form $\forall X(p(X) \wedge s(X) \rightarrow \bot)$ and a fixed query is already coNP-hard, which in turn implies the coNP-hardness result in Theorem 2. Actually, the latter is implicit [2, Example 5], and it can be shown by a reduction from a variant of UNSAT, called 2+2UNSAT, where each clause has two positive and two negative literals, where the literals involve either regular variables or the truth constant true or false.

4 From Classical to AR-CQ Answering

We now focus on the main decidable classes of TGDs, enriched with NCs, and we show that the complexity of AR-CQ answering can be obtained in a uniform way by exploiting our generic complexity theorems. Recall that the main (syntactic) conditions on TGDs that guarantee the decidability of CQ answering are the following: (1) guardedness [3], which guarantees the treelikeness of the underlying canonical models; (2)

	Combined	ba**-combined**	fp**-combined**	**Data**
Guarded	2EXPTIME	EXPTIME	NP	PTIME
Weakly-Guarded	2EXPTIME	EXPTIME	EXPTIME	EXPTIME
Sticky	EXPTIME	NP	NP	in AC_0
Weakly-Sticky	2EXPTIME	2EXPTIME	NP	PTIME
Acyclic	NEXPTIME	NEXPTIME	NP	in AC_0
Weakly-Acyclic	2EXPTIME	2EXPTIME	NP	PTIME

Table 1. CQ answering. All results are completeness results, unless stated otherwise.

	Combined	ba**-combined**	fp**-combined**	**Data**
Guarded	2EXPTIME	EXPTIME	Π_2^p	coNP
Weakly-Guarded	2EXPTIME	EXPTIME	EXPTIME	EXPTIME
Sticky	EXPTIME	Π_2^p	Π_2^p	coNP
Weakly-Sticky	2EXPTIME	2EXPTIME	Π_2^p	coNP
Acyclic	NEXP - P^{NE}	NEXP - P^{NE}	Π_2^p	coNP
Weakly-Acyclic	2EXPTIME	2EXPTIME	Π_2^p	coNP

Table 2. AR-CQ answering. A single complexity class in a cell refers to a completeness result, while two classes $\mathcal{C}_1$-$\mathcal{C}_2$ refer to $\mathcal{C}_1$-hardness and $\mathcal{C}_2$-membership.

stickiness [4], which ensures the termination of backward resolution; and (3) acyclicity, which guarantees the finiteness of the underlying canonical models. Interestingly, each one of the above conditions has its "weakly" counterpart: weak-guardedness [3], weak-stickiness [4] and weak-acyclicity [6], respectively. The complexity of CQ answering under the above classes of TGDs is summarized in Table 1. Clearly, Table 1 and Theorems 1 and 2 imply Table 2, apart from the (ba-)combined complexity for acyclic TGDs and NCs; let us briefly comment on this.

The (ba-)combined complexity of CQ answering under acyclic TGDs has to our knowledge never been explicitly studied; we show that is NEXPTIME-complete: the upper bound is obtained by a reduction to nonrecursive logic programming [5], while the lower bound by a reduction from a TILING problem [7]. Notice that Theorem 1 does not cover the cases where classical CQ answering is in a nondeterministic class above PSPACE. Nevertheless, by exploiting the guess-and-check algorithm discussed in the proof of Theorem 1, we obtain $\text{coNP}^{\text{NEXPTIME}}$ upper bound. It is implicit in [8] that $\text{NP}^{\text{NEXPTIME}} = \text{P}^{\text{NE}}$, and since P^{NE} is a deterministic class, $\text{coP}^{\text{NE}} = \text{P}^{\text{NE}}$. Consequently, AR-CQ answering under acyclic TGDs and NCs is in P^{NE} in (ba-)combined complexity; the NEXPTIME-hardness is inherited from classical query answering.

5 Conclusions

In this work, which is a short version of [10], we performed an in-depth complexity analysis of the problem of consistent query answering under the main decidable classes of TGDs, focussing on the AR semantics. Notably, generic complexity results have been established, which allowed us to obtain a (nearly) complete picture of the complexity of our problem in a systematic and uniform way. Regarding future work, apart from bridging the complexity gap for acyclic TGDs, we intend to perform a similar

complexity analysis for other important semantics such as the IAR semantics, that is, a sound approximation of the AR semantics [9].

Acknowledgements. This work has been funded by the EPSRC grant EP/J008346/1. M.V. Martinez and G.I. Simari are partially supported by Proyecto PIP-CONICET 112-201101-01000. A. Pieris is also supported by the Austrian Science Fund (FWF): P25207-N23 and Y698.

References

1. Arenas, M., Bertossi, L.E., Chomicki, J.: Consistent query answers in inconsistent databases. In: PODS. pp. 68–79 (1999)
2. Bienvenu, M.: On the complexity of consistent query answering in the presence of simple ontologies. In: AAAI (2012)
3. Calì, A., Gottlob, G., Kifer, M.: Taming the infinite chase: Query answering under expressive relational constraints. J. Artif. Intell. Res. 48, 115–174 (2013)
4. Calì, A., Gottlob, G., Pieris, A.: Towards more expressive ontology languages: The query answering problem. Artif. Intell. 193, 87–128 (2012)
5. Dantsin, E., Voronkov, A.: Complexity of query answering in logic databases with complex values. In: LFCS. pp. 56–66 (1997)
6. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: Semantics and query answering. Theor. Comput. Sci. 336(1), 89–124 (2005)
7. Fürer, M.: The computational complexity of the unconstrained limited domino problem (with implications for logical decision problems). In: Logic and Machines. pp. 312–319 (1983)
8. Hemachandra, L.A.: The strong exponential hierarchy collapses. J. Comput. Syst. Sci. 39(3), 299–322 (1989)
9. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: RR. pp. 103–117 (2010)
10. Lukasiewicz, T., Martinez, M.V., Pieris, A., Simari, G.I.: From classical to consistent query answering under existential rules. In: AAAI (2015)
11. Rosati, R.: On the complexity of dealing with inconsistency in description logic ontologies. In: IJCAI. pp. 1057–1062 (2011)