

Querying Inconsistent Description Logic Knowledge Bases under Preferred Repair Semantics (Extended Abstract)

Meghyn Bienvenu¹, Camille Bourgaux¹, and François Goasdoué²

¹LRI, CNRS & Université Paris-Sud, Orsay, France

²IRISA, Université de Rennes 1, Lannion, France

Abstract. Many inconsistency-tolerant semantics for DLs rely on the notion of a repair, i.e., a \subseteq -maximal subset of the ABox that is consistent with the TBox. This extended abstract presents our investigation [4] of variants of two popular inconsistency-tolerant semantics obtained by replacing classical repairs by various types of preferred repair.

1 Introduction

Recent years have seen an increasing interest in performing database-style query answering over description logic (DL) knowledge bases (KBs). Since scalability is crucial in data-rich applications, there has been a trend to using lightweight DLs for which query answering is tractable w.r.t. the size of the ABox. Particular attention has been paid to DLs of the DL-Lite family [6] for which query answering can be reduced to evaluation of standard database queries.

An important issue that arises in the context of DL query answering is how to handle the case in which the ABox is inconsistent with the TBox. Indeed, while it may be reasonable to assume that the TBox has been properly debugged, the ABox will typically be very large and subject to frequent modifications, both of which make errors likely. Since it may be too difficult or costly to identify and fix these errors, it is essential to be able to provide meaningful answers to queries in the presence of such data inconsistencies. To address this issue, several different inconsistency-tolerant semantics have been proposed for querying inconsistent DL knowledge bases. Among them, the AR and IAR semantics [10] are the most well-known and well-studied. Both semantics are based upon the notion of a *repair*, defined as an inclusion-maximal subset of the ABox which is consistent with the TBox. The AR semantics, which was inspired by consistent query answering in relational databases (cf. [2] for a survey), amounts to computing those answers that hold no matter which repair is chosen, whereas the more cautious IAR semantics queries the intersection of the repairs.

When additional information on the reliability of ABox assertions is available, it is natural to use this information to identify *preferred repairs*, and to use the latter as the basis of inconsistency-tolerant query answering. This motivated us to study variants of the AR and IAR semantics obtained by replacing the classical

notion of repair by one of four different types of preferred repairs. Cardinality-maximal repairs, denoted \leq -repairs, are intended for settings in which all ABox assertions are believed to have the same likelihood of being correct. The other three types of preferred repairs target the scenario in which some assertions are considered to be more reliable than others, which can be captured qualitatively by partitioning the ABox into *priority levels* (and then applying either the set inclusion (\subseteq_P -repairs) or cardinality (\leq_P -repairs) criterion to each level), or quantitatively by assigning *weights* to the ABox assertions and selecting those repairs having the greatest weight (\leq_w -repairs).

2 Complexity Results

There have been several studies of the complexity of querying DL knowledge bases under the AR and IAR semantics for the standard notion of repairs [10, 14, 3, 5]. The complexity of AR query answering with \leq - or \leq_w -repairs was first studied in the database setting for denial constraints [11], and more recently in the DL setting for the highly expressive *SHIQ* [9]. However, there had been no systematic study of the computational properties of preferred repairs for important lightweight DLs. We addressed this gap in the literature by studying the complexity of answering conjunctive and atomic queries under the eight preferred repair semantics (cf. Figure 1). We focused on the lightweight logic DL-Lite \mathcal{R} (which underlies the OWL 2 QL profile [13]), though many of our results hold also for other data-tractable ontology languages.

Unsurprisingly, the results are mainly negative, showing that adding preferences increases complexity. For the IAR semantics, we move from polynomial data complexity in the case of (plain) IAR semantics to coNP-hard data complexity (or worse) for IAR semantics based on preferred repairs. For the AR semantics, query answering is known to be coNP-complete in data complexity already for the standard notion of repairs, but adding preferences often results in even higher complexities. The sole exception is \subseteq_P -repairs (which combine priority levels and the set inclusion criterion), for which both AR and IAR query answering are “only” coNP-complete in data complexity.

The lower combined complexity of the IAR semantics shows that they still retain some computational advantage over the AR semantics. Intuitively, this is because one can precompute the intersection of repairs in an offline phase and then utilize standard querying algorithms at query time, whereas no such pre-computation is possible for the AR semantics.

3 System and Experiments

We are aware of two systems for inconsistency-tolerant query answering over DL KBs: the system of [9] for querying *SHIQ* KBs under \leq_w -AR semantics, and the QuID system [15] that handles IAR semantics (without preferences) and DL-Lite \mathcal{A} KBs. The latter work left open whether, for the standard notion of repairs,

	AR	IAR		AR	IAR
\subseteq	coNP	in P	\subseteq	Π_2^P	NP
\leq	$\Delta_2^P[\mathcal{O}(\log n)]$	$\Delta_2^P[\mathcal{O}(\log n)]$	\leq	Π_2^P	$\Delta_2^P[\mathcal{O}(\log n)]$
\subseteq_P	coNP	coNP	\subseteq_P	Π_2^P	$\Delta_2^P[\mathcal{O}(\log n)]$
\leq_P & \leq_w	$\Delta_2^{P\dagger}$	$\Delta_2^{P\dagger}$	\leq_P & \leq_w	Π_2^P	$\Delta_2^{P\dagger}$

Data complexity Combined complexity

Fig. 1: Complexity of conjunctive query (CQ) entailment over DL-Lite KBs under AR and IAR semantics for different types of preferred repairs. For atomic queries, the data and combined complexity coincide with the data complexity for CQs. All results are completeness results unless otherwise noted. New results in bold. $\dagger \Delta_2^P[\mathcal{O}(\log n)]$ -complete if there is a bound on the number of priority classes (resp. maximal weight).

the IAR semantics constitutes a good approximation of the AR semantics, and whether the AR semantics could be feasibly computed in practice.

Encouraged by the performance of modern-day SAT solvers, we proposed a practical SAT-based approach for query answering in DL-Lite \mathcal{R} under the (plain) AR, \subseteq_P -IAR, and \subseteq_P -AR semantics. We showed how to encode query answering under these three semantics in terms of propositional unsatisfiability, using a reachability analysis on the conflict graph (inspired by [7]) to reduce the size of the encodings. We implemented the approach in our CQAPri system. The ABox is stored in a PostgreSQL server, while the TBox is kept in-memory together with the pre-computed set of conflicts for the KB. When a query arrives, CQAPri evaluates it over the ABox using its SQLized rewriting (obtained with the Rapid query rewriting engine [8]), to get its *possible answers* and their causes. Among the possible answers, CQAPri identifies the IAR ones, or an approximation of the \subseteq_P -IAR ones. For those possible answers that are not found to be IAR answers (resp. in the approximation of the \subseteq_P -IAR answers), deciding whether they are entailed under the AR semantics (resp. the \subseteq_P -IAR and \subseteq_P -AR semantics), is done using our encodings and the SAT4J SAT solver [1].

An experimental evaluation demonstrates the scalability of the approach in settings we presume realistic. We considered the DL-Lite \mathcal{R} LUBM $_{20}^{\exists}$ TBox from [12] and the associated Extended University Data Generator (EUDG). We extended LUBM $_{20}^{\exists}$ with negative inclusions and introduced inconsistencies in the ABox. Prioritizations of ABox were made to capture a variety of scenarios. The experiments show that our system scales up to large ABoxes (up to about 2 million assertions) for the IAR/AR and \subseteq_P -IAR/ \subseteq_P -AR semantics, when the number of conflicting assertions varies from a few percent (for all of these semantics) to a few tens of percent (only for IAR/AR). This positive empirical result is due in large part to the efficacy of the incomplete methods, which leave only few cases to be handled by the SAT solver. In a similar vein, our simple approximation of the \subseteq_P -IAR semantics often produced a large share of the \subseteq_P -IAR answers, which themselves constituted a large portion of the \subseteq_P -AR answers.

Acknowledgements This work was supported by contract ANR-12-JS02-007-01. We acknowledge Despoina Trivela for her work on an early version of the system.

References

1. Berre, D.L., Parrain, A.: The sat4j library, release 2.2. *JSAT* 7(2-3), 59–64 (2010)
2. Bertossi, L.E.: Database Repairing and Consistent Query Answering. Synthesis Lectures on Data Management, Morgan & Claypool Publishers (2011)
3. Bienvenu, M.: On the complexity of consistent query answering in the presence of simple ontologies. In: Proc. of AAAI (2012)
4. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Querying inconsistent description logic knowledge bases under preferred repair semantics. In: Proc. of AAAI (2014)
5. Bienvenu, M., Rosati, R.: Tractable approximations of consistent query answering for robust ontology-based data access. In: Proc. of IJCAI (2013)
6. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automated Reasoning* 39(3), 385–429 (2007)
7. Chomicki, J., Marcinkowski, J., Staworko, S.: Computing consistent query answers using conflict hypergraphs. In: Proc. of CIKM. pp. 417–426 (2004)
8. Chortaras, A., Trivela, D., Stamou, G.: Optimized query rewriting for OWL 2 QL. In: Proc. of CADE (2011)
9. Du, J., Qi, G., Shen, Y.D.: Weight-based consistent query answering over inconsistent *SHIQ* knowledge bases. *Knowledge and Information Systems* 34(2), 335–371 (2013)
10. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Proc. of RR. pp. 103–117 (2010)
11. Lopatenko, A., Bertossi, L.E.: Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics. In: Proc. of ICDT. pp. 179–193 (2007)
12. Lutz, C., Seylan, I., Toman, D., Wolter, F.: The combined approach to OBDA: Taming role hierarchies using filters. In: Proc. of ISWC. pp. 314–330 (2013)
13. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language profiles. W3C Recommendation (11 December 2012), <http://www.w3.org/TR/owl2-profiles/>, available at <http://www.w3.org/TR/owl2-profiles/>
14. Rosati, R.: On the complexity of dealing with inconsistency in description logic ontologies. In: Proc. of IJCAI. pp. 1057–1062 (2011)
15. Rosati, R., Ruzzi, M., Graziosi, M., Masotti, G.: Evaluation of techniques for inconsistency handling in OWL 2 QL ontologies. In: Proc. of ISWC. pp. 337–349 (2012)