Querying Inconsistent Prioritized Data with ORBITS: Algorithms, Implementation, and Experiments

Extended Abstract

Meghyn Bienvenu[1], Camille Bourgaux[2]

[1]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, Talence, France
[2]DI ENS, ENS, CNRS, PSL University & Inria, Paris, France

Abstract

This extended abstract of [1] presents our investigation of practical algorithms for inconsistency-tolerant query answering over prioritized knowledge bases. We introduce SAT encodings for Pareto- and completion-optimal repairs w.r.t. general priority relations over the knowledge base facts and propose several ways of computing answers under (optimal) repair-based semantics by exploiting different reasoning modes of SAT solvers. Proofs, pseudo-code for algorithms, and details on the experimental evaluation are provided in the appendix of [2].

Keywords

inconsistency-tolerant semantics, prioritized knowledge bases, SAT-based algorithms

1. Querying Inconsistent Prioritized Knowledge Bases

When a knowledge base (KB) consisting of a dataset and a logical theory (be it an ontology or a set of database dependencies) is such that the data is inconsistent with the constraints, a prominent approach is to adopt inconsistency-tolerant semantics in order to extract meaningful information from the contradictory data. In the database setting, such an approach goes by the name of *consistent query answering (CQA)* and has been extensively studied [3, 4]. A central notion is that of a *(subset) repair*, defined as a maximal subset of the dataset that satisfies the constraints. Intuitively, repairs represent all different ways of minimally modifying the data to satisfy the constraints. As we do not know which repair corresponds to the true part of the data, the CQA semantics stipulates that a tuple is a query answer if it is an answer w.r.t. *every repair*. Inconsistency-tolerant semantics have also drawn considerable interest in the setting of ontology-mediated query answering (OMQA) [5, 6, 7]. In addition to the *AR semantics* (the OMQA analog of the CQA semantics), several other inconsistency-tolerant semantics have been proposed (see [8, 9] for surveys and references), among which: the *brave semantics* [10], which only requires a tuple to be an answer w.r.t. *some repair*, provides a natural notion of possible answer, and the *IAR semantics* [11], which answers queries over the *intersection of the repairs*, identifies the most reliable answers.

DL 2022: 35th International Workshop on Description Logics, August 7–10, 2022, Haifa, Israel

meghyn.bienvenu@labri.fr (M. Bienvenu); camille.bourgaux@ens.fr (C. Bourgaux)
0000-0001-6229-8103 (M. Bienvenu); 0000-0002-8806-6682 (C. Bourgaux)

The basic notion of repair can be refined by exploiting preference information. An approach introduced in the database setting [12] and recently explored in the OMQA setting [13] assumes that preferences are given by a binary *priority relation* between conflicting facts. Three notions of 'best' repairs w.r.t. a priority relation were proposed, namely, Pareto-optimal, globally-optimal, and completion-optimal repairs, and can be used in place of subset repairs in any repair-based semantics. In the case where the priority relation is *score-structured*, that is, induced by assigning scores to facts, the three kinds of optimal repair coincide.

The complexity of answering queries under (optimal) repair-based semantics has been extensively studied in the database and OMQA settings, refer to [4, 8] for an overview and references. We can briefly summarize these (many!) complexity results as follows: query answering under the AR (or CQA) semantics is coNP-hard in data complexity even in the simplest of settings (e.g. key constraints, class disjointness), and adopting optimal repairs in place of subset repairs leads to (co)NP-hardness for the brave and IAR semantics as well. Membership in (co)NP holds for AR, brave, and IAR semantics w.r.t. subset, Pareto-optimal, and completion-optimal repairs in the most commonly considered settings i.e. for database constraints given by primary keys or more generally, functional dependencies (FDs), and for ontologies formulated in data-tractable description logics such as those of the DL-Lite family [14]. Globally-optimal repairs lead to higher complexity and are thus not considered in this paper.

The preceding (co)NP complexity results naturally suggest the interest of employing SAT solvers. Two recent systems, CQAPri [15, 16] – which targets DL-Lite KBs and AR, brave, and IAR semantics, w.r.t. subset repairs as well as optimal repairs for the restricted class of score-structured priority relations – and CAvSAT [17] – which targets relational databases and AR semantics w.r.t. subset repairs – have begun to explore such an approach. While geared to different forms of constraints, the two systems solve essentially the same problem, yet they employ SAT solvers in different ways. This motivates a comprehensive study of the use of SAT-based approaches for inconsistency-tolerant query answering, which abstracts from the particular setting and provides a solid foundation for the future development of such systems.

2. SAT-Based Algorithms

We propose SAT-based algorithms to answer queries under the semantics that fall in the (co)NP complexity class: X-AR, X-brave and X-IAR where X $\in$ {S,P,C} indicates the kind of repair: subset, Pareto-optimal, and completion-optimal respectively. They rely on pre-computed *conflicts*, defined as the minimal inconsistent subsets of the data, and *causes* for a query *potential answer*, defined as the minimal consistent subsets of the data that entails the query answer together with the logical theory. While our algorithms can be applied to any KB for which we can compute the conflicts and causes, the overall complexity of the resulting query answering algorithms depends on the cost of computing these inputs. For FO-rewritable ontology languages (like DL-Lite) or databases equipped with denial constraints, the sets of conflicts, candidate answers, and their causes, can be computed in PTime via database query evaluation, yielding procedures of the expected (co)NP complexity. We focus on the case where conflicts are *binary* but we discuss how to extend the encodings to the general case. Binary conflicts allow us to define a graph representation of conflicts and priorities: The *directed conflict graph* has facts as nodes and an edge from α to β iff α and β are in conflict and α is not preferred to β.

We provide propositional encodings of the X-AR, X-brave, and X-IAR semantics, including the first encodings for Pareto- and completion-optimal repairs. Our encodings are generic and are built in a modular manner from a core set of basic formulas, some of them for which we consider several variants. In particular, we consider two ways of encoding the absence, or contradiction, of a cause for the query, and two ways of encoding Pareto-optimal repair maximality. In the case of score-structured priority relations, since Pareto-optimal and completion-optimal repairs coincide, this gives three possible encodings of maximality. For each semantics, we provide several encodings, that handle either a single potential answer or several answers at the same time. For the X-brave and X-IAR semantics we additionally provide encodings to check whether a given cause is in some or every optimal repair, or if some fact is in all optimal repairs.

Based upon these encodings, we develop several algorithms which utilize different functionalities of modern SAT solvers. An initial preprocessing step serves to (1) handle self-inconsistent facts, and (2) find the answers that have some cause that contains only facts without any outgoing edge in the directed conflict graph, and thus trivially hold in all optimal repairs. It then remains to filter the remaining potential answers. The four first algorithms we propose to do so are generic in the sense that they can be used for all semantics.

- The first one is similar to the algorithm used by CQAPri: For each answer to filter, it checks whether the corresponding SAT encoding is satisfiable.
- The second one is similar to the CAvSAT algorithm: It handles all potential answers together with a weighted MaxSAT instance where soft clauses correspond to answers.
- The third one uses the same multi-answer encoding and relies on minimal unsatisfiable subsets of the soft clauses w.r.t. the hard clauses to filter the answers.
- The fourth one iteratively evaluates the multi-answer encoding, treating the variables corresponding to potential answers as assumptions.

While we may need to consider all causes to decide whether an answer holds under X-AR semantics, in the X-brave or X-IAR case it is sufficient to find a single cause that belongs to some or every optimal repair. We hence propose algorithms specific to these cases.

- The first one checks for each cause whether it belongs to some/every optimal repair using the dedicated encoding.
- The second one is specific to X-IAR. It considers the answers and their causes in turn while maintaining two sets of facts, checking facts individually as it goes: the X-IAR facts that belong to the intersection of the optimal repairs and the non-X-IAR facts.
- The last one is also specific to X-IAR. The difference with the previous one is that for each answer, it uses a weighted MaxSAT solver to decide which facts hold under X-IAR among those that belong to some cause and have not already been checked.

3. Implementation and Experiments

We implemented the proposed algorithms in our ORBITS[1] system (Optimal Repair-Based Inconsistency-Tolerant Semantics). ORBITS takes as input two JSON files containing respectively

[1] https://github.com/bourgaux/orbits

the directed conflict graph and the potential answers associated to their causes. The user also specifies a semantics (AR, IAR, or brave), a kind of repair together with the encoding variants to use to encode optimality and contradiction, and the algorithm to use to compute the answers w.r.t. the chosen semantics. The set of answers is output as a JSON file.

We evaluated ORBITS on three (sets of) KBs. The first is the CQAPri benchmark [18], a synthetic benchmark crafted to evaluate inconsistency-tolerant query answering over DL-Lite KBs, adapted from the LUBM$^{\exists}_{20}$ benchmark [19]. The two others, called Food Inspection and Physicians, are real-world datasets built from public open data [20, 21, 22], which have already been used to evaluate data cleaning and consistent query answering systems [23, 17]. They consist of relational databases built from the original csv files, on which typical integrity constraints (keys and FDs) have been added. The size of the conflict graphs we obtain ranges from 2K to 946K facts and 2K to 3M conflicts. We added score-structured and non-score-structured priority relations on these conflict graphs.

Our experimental evaluation aims at assessing (i) the impact of adopting different kinds of repairs, and (ii) the relative performances of alternative procedures for the same semantics. More precisely, we consider the following questions.

- What is the impact in terms of number of answers of adopting optimal repairs rather than standard repairs, or completion-optimal repairs instead of Pareto-optimal repairs when the priority relation is not score-structured?
- How do different kinds of repairs compare in terms of computation time?
- Given a semantics and type of repair, what is the impact in terms of computation times of the choice of: How to encode optimality ? How to encode contradictions ? The algorithm used to filter the non-trivial answers?

Our most important finding is that the choice of an algorithm and encoding can have a huge impact on the computation time: Changing a single parameter among the algorithm, optimality encoding, and contradiction encoding can result in a significant change (sometimes of several orders of magnitude). The comparison of the possible procedures for each semantics on the different datasets and queries shows that there is not a ‘best’ method in general. However, we still gain some relevant insights. For example we found that one of the three optimality encodings often performs better while the one based on completion-optimal repairs never significantly outperforms the others. We also found that one of the algorithms specifically tailored for the X-IAR semantics is generally the best one to use with this semantics. Finally, we observe that one variant of the contradiction encoding does not work well with one variant of the optimality encoding in general.

While in some cases our results can be used to single out some approaches as more effective, more often there are no clear winner(s). This suggests that to minimize runtimes, it may make sense to launch multiple algorithms in parallel, and/or devise methods that can help predict which algorithm and encoding will perform best on a given dataset and query, e.g. using machine learning techniques.

Acknowledgements

This work was supported by the ANR AI Chair INTENDED (ANR-19-CHIA-0014).

References

[1] M. Bienvenu, C. Bourgaux, Querying inconsistent prioritized data with ORBITS: Algorithms, implementation, and experiments, in: Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR) (to appear), 2022.

[2] M. Bienvenu, C. Bourgaux, Querying inconsistent prioritized data with ORBITS: Algorithms, implementation, and experiments, 2022. arxiv.org/abs/2202.07980 [cs.LO].

[3] M. Arenas, L. E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: Proceedings of the 18th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS), 1999, pp. 68–79.

[4] J. Wijsen, Foundations of query answering on inconsistent databases, SIGMOD Record 48 (2019) 6–16. URL: https://doi.org/10.1145/3377391.3377393. doi:10.1145/3377391.3377393.

[5] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, Journal of Data Semantics 10 (2008) 133–173.

[6] M. Bienvenu, M. Ortiz, Ontology-mediated query answering with data-tractable description logics, in: Tutorial Lectures of the 11th Reasoning Web International Summer School, 2015, pp. 218–307.

[7] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, M. Zakharyaschev, Ontology-based data access: A survey, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 5511–5519.

[8] M. Bienvenu, C. Bourgaux, Inconsistency-tolerant querying of description logic knowledge bases, in: Tutorial Lectures of the 12th International Reasoning Web Summer School, 2016, pp. 156–202.

[9] M. Bienvenu, A short survey on inconsistency handling in ontology-mediated query answering, Künstliche Intelligenz 34 (2020) 443–451. URL: https://doi.org/10.1007/s13218-020-00680-9. doi:10.1007/s13218-020-00680-9.

[10] M. Bienvenu, R. Rosati, Tractable approximations of consistent query answering for robust ontology-based data access, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), 2013.

[11] D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, D. F. Savo, Inconsistency-tolerant semantics for description logics, in: Proceedings of the 4th International Conference on Web Reasoning and Rule Systems (RR), 2010, pp. 103–117.

[12] S. Staworko, J. Chomicki, J. Marcinkowski, Prioritized repairing and consistent query answering in relational databases, Annals of Mathematics and Artifcial Intelligence (AMAI) 64 (2012) 209–246. URL: https://doi.org/10.1007/s10472-012-9288-8. doi:10.1007/s10472-012-9288-8.

[13] M. Bienvenu, C. Bourgaux, Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation, in: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR), 2020, pp. 141–151.

[14] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The DL-Lite family, Journal of Automated

Reasoning (JAR) 39 (2007) 385–429. URL: https://doi.org/10.1007/s10817-007-9078-x. doi:10.1007/s10817-007-9078-x.

[15] M. Bienvenu, C. Bourgaux, F. Goasdoué, Querying inconsistent description logic knowledge bases under preferred repair semantics, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI), 2014, pp. 996–1002.

[16] M. Bienvenu, C. Bourgaux, F. Goasdoué, Computing and explaining query answers over inconsistent DL-Lite knowledge bases, Journal of Artificial Intelligence Research (JAIR) 64 (2019) 563–644. URL: https://doi.org/10.1613/jair.1.11395. doi:10.1613/jair.1.11395.

[17] A. A. Dixit, P. G. Kolaitis, A SAT-based system for consistent query answering, in: Proceedings of the 22nd International Conference on Theory and Applications of Satisfiability Testing (SAT), 2019, pp. 117–135.

[18] C. Bourgaux, Inconsistency Handling in Ontology-Mediated Query Answering. (Gestion des incohérences pour l'accès aux données en présence d'ontologies), Ph.D. thesis, University of Paris-Saclay, France, 2016. URL: https://tel.archives-ouvertes.fr/tel-01378723.

[19] C. Lutz, I. Seylan, D. Toman, F. Wolter, The combined approach to OBDA: Taming role hierarchies using filters, in: Proceedings of the 12th International Semantic Web Conference (ISWC), 2013, pp. 314–330.

[20] Dataset: Food Inspections, Chicago Data Portal, https://data.cityofchicago.org/Health-Human-Services/Food-Inspections/4ijn-s7e5, accessed December 7, 2020.

[21] Dataset: New York City Restaurant Inspection Results, Department of Health and Mental Hygiene (DOHMH), NYC Open Data, https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/43nn-pn8j, accessed December 7, 2020.

[22] Dataset: National Downloadable File, Centers for Medicare & Medicaid Services, https://data.cms.gov/provider-data/dataset/mj5m-pzi6, accessed December 10, 2020.

[23] T. Rekatsinas, X. Chu, I. F. Ilyas, C. Ré, Holoclean: Holistic data repairs with probabilistic inference, Proceedings of the VLDB Endowment (PVLDB) 10 (2017) 1190–1201.