HAL
archives-ouvertes.fr

Ontology-Mediated Query Answering: Harnessing Knowledge to Get More From Data

Meghyn Bienvenu

▶ To cite this version:

Meghyn Bienvenu. Ontology-Mediated Query Answering: Harnessing Knowledge to Get More From Data. IJCAI: International Joint Conference on Artificial Intelligence, Jul 2016, New York, United States. 25th International Joint Conference on Artificial Intelligence, 2016, <http://ijcai-16.org/>. <lirmm-01367866>

HAL Id: lirmm-01367866
https://hal-lirmm.ccsd.cnrs.fr/lirmm-01367866

Submitted on 16 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ontology-Mediated Query Answering: Harnessing Knowledge to Get More From Data

Meghyn Bienvenu
CNRS, Université de Montpellier, INRIA
Montpellier, France

Abstract

Ontology-mediated query answering (OMQA) is a new paradigm in data management that seeks to exploit the semantic knowledge expressed in ontologies to improve query answering over data. This paper briefly introduces OMQA and gives an overview of two recent lines of research.

1 Introduction

In recent years, there has been growing interest both from academia and from industry in ontology-mediated query answering (OMQA), in which the semantic knowledge provided by ontologies is used to improve query answering. Ontologies are used to enrich the vocabulary of data sources, allowing users to formulate their queries in a more familiar vocabulary which abstracts from the specific way data is stored. In information integration, ontologies serve to relate the vocabularies of different data sources and to provide a unified view to the user. Finally, ontologies help tackle data incompleteness by allowing inference of new facts from the ontology and the data, providing a more complete set of query results.

To illustrate the above points, let us consider the application of OMQA techniques in medicine. Each hospital will store its patient data in databases, using terms from a standardized medical ontology (or linking to these terms via mappings). The information in patient records will typically use specialized terms, for instance, stating that a patient suffers from Hodgkin's lymphoma and has been prescribed a drug for hypertension. A hospital admin may query the system to find cancer patients being treated for high-blood pressure. In the absence of an ontology, this query is likely to turn up no results, as the generic terms 'cancer' and 'high blood pressure' do not explicitly appear in patient records. However, the ontology contains a hierarchy of terms, from highly specialized to generic, and the semantic relationships between them. OMQA systems will perform the necessary inferences (e.g. inferring that Hodgkin's lymphoma is a type of cancer) in order to obtain all (deducible) answers to the query. Moreover, by utilizing a standardized ontology, patient data from different hospitals can be seamlessly integrated.

Description logics (DLs) are among the most commonly used and well-studied ontology languages. Expressive DLs (like $\mathcal{ALC}$ and $\mathcal{SHIQ}$) allow fine-grained modeling and provide the logical underpinnings for the W3C-standardized OWL web ontology language. Horn DLs like $\mathcal{EL}$ and its extensions offer better computational properties by forbidding (implicit or explicit) use of disjunction, and they are a popular choice for formalizing knowledge in medicine and the life sciences; the OWL 2 EL profile is based upon such DLs. The DL-Lite family of DLs (and corresponding OWL 2 QL profile) were specifically designed with OMQA in mind. To scale up to large datasets, these languages are quite restricted in expressivity, but are still able to capture key modeling constructs, as illustrated in the next example:

Example 1. Here are some DL-Lite axioms about academia:

$$(1)\quad \mathsf{Prof} \sqsubseteq \mathsf{Faculty} \qquad (2)\quad \mathsf{Fellow} \sqsubseteq \mathsf{Faculty}$$
$$(3)\quad \mathsf{Prof} \sqsubseteq \neg\mathsf{Fellow} \qquad (4)\quad \mathsf{Prof} \sqsubseteq \exists\mathsf{Teaches}$$
$$(5)\quad \exists\mathsf{Teaches} \sqsubseteq \mathsf{Faculty} \qquad (6)\quad \exists\mathsf{Teaches}^{-} \sqsubseteq \mathsf{Course}$$

Axioms (1)-(2) express that professors and research fellows are both types of faculty members, while (3) states that the two classes are disjoint. Axiom (4) requires that every professor teach at least one course. The final two axioms express that the relation Teaches links faculty members to courses.

The most commonly used queries in OMQA are conjunctive queries (CQs), which are built from atoms using conjunction and existential quantifiers, and correspond to the SPJ fragment of SQL and basic graph patterns in SPARQL.

Example 2. The following CQ can be used to find faculty members that teach some course:

$$q_1(x) = \exists y.\mathsf{Faculty}(x) \wedge \mathsf{Teaches}(x, y)$$

If we use an OMQA system to answer q_1 over the dataset

$$\mathcal{D}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

using axioms (1)-(6) as our ontology, then both anna and tom will be returned as answers. This is because from (1), (4),and $\mathsf{Prof}(\mathsf{anna})$, we can infer that anna is faculty and teaches, and from (2) and $\mathsf{Fellow}(\mathsf{tom})$, we can derive $\mathsf{Faculty}(\mathsf{anna})$.

To give a flavour for OMQA research, this short paper provides an overview of two recent lines of research to which the author has contributed. Section 2 is concerned with understanding the limits and possibilities of query rewriting, a key algorithmic technique for OMQA, while Section 3 tackles the issue of making OMQA robust to data inconsistencies. Section 4 provides a brief discussion of the future of OMQA research and pointers to the literature.

2 Query Rewriting: Limits and Possibilities

Query rewriting represents one of the most promising algorithmic approaches to ontology-mediated query answering. The approach consists of a rewriting step in which the input query (typically, a conjunctive query) is transformed into a new database query (called a *rewriting*) which encodes the relevant information from the ontology, followed by a second step in which the rewriting is evaluated over the data using a database system. Thus, query rewriting reduces the OMQA problem to the simpler problem of database query evaluation, thereby allowing one to take advantage of the efficiency and maturity of database systems. In particular, if one requires that the rewriting is a first-order (FO) query, then rewritings can be rephrased in SQL and evaluated using highly-optimized relational database management systems.

Example 3. Consider the DL-Lite ontology $\mathcal{O}$ consisting of axioms (1)-(4). The following query

$$(\mathsf{Faculty}(x) \vee \mathsf{Fellow}(x)) \wedge \exists y.\mathsf{Teaches}(x, y)) \vee \mathsf{Prof}(x)$$

is a rewriting of the query q_1 w.r.t. $\mathcal{O}$. If we evaluate the preceding query over the dataset $\mathcal{D}_1$, we obtain both anna and tom as answers, as expected.

Succinctness and Optimality of Rewritings

The DL-Lite family of DLs [Calvanese *et al.*, 2007] (and corresponding OWL 2 QL profile) were specifically designed to ensure the existence of FO-rewritings for all conjunctive queries, and several query rewriting algorithms have been developed and implemented for these languages. However, experimental evaluation showed that the rewritings produced by such rewriting engines were often huge, making them difficult, or even impossible, to evaluate. When rewritings are given as union of conjunctive queries (UCQs), which is the case for many rewriting algorithms, it is not difficult to show that such rewritings can be exponentially large.

A natural question is whether an exponential blowup can be avoided if rewritings are expressed in richer query languages, like PE-queries, non-recursive datalog (NDL) queries, or full FO-queries. More generally, under what conditions can we ensure polynomial-size rewritings? For ontologies expressed in DL-Lite$_\mathcal{R}$ (or OWL 2 QL), a first answer was given in [Kikot *et al.*, 2012], which proved exponential lower bounds for the worst-case size of PE- and NDL-rewritings, as well as a superpolynomial lower bound for FO-rewritings (assuming NP $\not\subseteq$ P/poly). These initial negative results spurred a systematic parameterized study [Kikot *et al.*, 2014; Bienvenu *et al.*, 2015; 2016d] of the impact of restricting the query shape (linear, tree-shaped, bounded treewidth) and/or ontology depth on the size of rewritings. The resulting succinctness landscape shows that even in very restricted settings (namely, linear queries and depth 2 ontologies), PE-rewritings may be of super-polynomial size, whereas, for many combinations of queries and ontologies (in particular, for bounded treewidth queries coupled with bounded depth ontologies), polynomial-size NDL-rewritings are guaranteed to exist. These results, which were obtained by establishing unexpectedly tight connections between query rewriting and circuit complexity, provide strong evidence in favour of adopting NDL as the target language for rewriting algorithms.

The preceding succinctness results have been complemented by a corresponding set of complexity results [Bienvenu *et al.*, 2015; 2016d]. It turns out that the structural classes of ontology-query pairs for which query answering is tractable (more precisely: NL- or LOGCFL-complete) are also classes admitting polysize NDL-rewritings. Rewriting-based query answering algorithms that achieve optimal worst-case complexity for these well-behaved classes have been recently proposed in [Bienvenu *et al.*, 2016e].

We note that all of the results discussed so far apply to ontologies formulated in DL-Lite$_\mathcal{R}$. For the DL-Lite$_{core}$ dialect (obtained by disallowing inclusions between binary relations), better results can sometimes be achieved [Bienvenu *et al.*, 2013b], but much work remains to obtain a complete picture of the succinctness landscape in DL-Lite$_{core}$.

Existence of Rewritings

At first glance, the FO query rewriting approach appears to have limited applicability, since for almost every DL outside the DL-Lite family (in particular, for $\mathcal{EL}$ and its extensions), we run into the problem that rewritings are not guaranteed to exist. However, such negative results reflect the worst-case situation and leave open the possibility that some, perhaps many, queries encountered in real applications are in fact FO-rewritable (and hence relational database technology can be used to answer such queries). Thus, an interesting and potentially quite useful research direction is to develop methods for identifying the cases where FO-rewriting is possible and to produce such rewritings when they exist.

A first step in this direction was made by [Bienvenu *et al.*, 2013a], who established decidability and complexity results for FO-rewritability of AQs in the presence of Horn DL ontologies, showing the problem to be EXPTIME-complete for DLs ranging from basic $\mathcal{EL}$ to the much more expressive Horn-$\mathcal{SHI}$. While these results were quite positive (similar problems in databases are known to be undecidable), the automata-based decision procedures used to show the upper bounds were ill-suited for implementation. However, by combining these theoretical results with an existing backward-chaining rewriting procedure, an efficent algorithm for testing FO-rewritability of atomic queries w.r.t. ontologies in $\mathcal{ELH}^{dr}$ (the basis for OWL 2 EL) was obtained [Hansen *et al.*, 2015]. Experimental results on real-world ontologies are very encouraging: the vast majority of AQs do possess FO-rewritings, and the computed rewritings (represented as NDL programs) are typically quite small.

A serious limitation of the preceding results is that they concern AQs, while CQs are required in many applications. A recent work [Bienvenu *et al.*, 2016c] addresses this gap by providing decision procedures and complexity results (ranging from EXPTIME- to 2EXPTIME-complete) for testing FO-rewritability of CQs in various Horn DLs. The next step will be to exploit these results to develop practical FO-rewritability algorithms for CQs, as was done for AQs.

For expressive DLs like $\mathcal{ALC}$, the first decision procedures for FO- and Datalog-rewritability of AQs were provided in [Bienvenu *et al.*, 2014b]. The latter work studied the expressive power of OMQA and established a surprising connection to constraint satisfaction problems (CSPs), which enabled the

transfer of deep results on FO- and Datalog-expressibility of CSPs to OMQA. Here also the design of practical algorithms for identifying rewritable queries and producing the corresponding rewritings constitutes an important challenge.

3 Inconsistency Handling in OMQA

In applications involving large datasets or multiple data sources, it is very likely that the data will be inconsistent with the ontology, rendering standard querying algorithms useless (as everything is entailed from a contradiction). Appropriate mechanisms for dealing with inconsistent data are thus crucial to the successful use of OMQA in practice.

Ideally, one would restore consistency by identifying and correcting the errors, but when this is not possible, a sensible strategy is to adopt an inconsistency-tolerant semantics which allows reasonable answers to be obtained despite the inconsistencies. The most well-known, and arguably the most natural, such semantics is the *AR semantics* [Lembo and Ruzzi, 2007], inspired by work on consistent query answering in databases. The semantics is based upon the notion of a *repair*, defined as an inclusion-maximal subset of the data that is consistent with the ontology. Intuitively, repairs capture the different ways of achieving consistency while retaining as much of the original data as possible. Query answering under AR semantics amounts to computing those query answers that hold for *every* repair (under standard semantics):

Example 4. Let $\mathcal{O}$ be as before, and let $\mathcal{D}_2 = \mathcal{D}_1 \cup \{\mathsf{Prof}(\mathsf{tom})\}$. The dataset $\mathcal{D}_2$ is inconsistent w.r.t. $\mathcal{O}$, as it violates axiom (3). There are two repairs of $\mathcal{D}_2$ w.r.t. $\mathcal{O}$:

$$\mathcal{R}_1 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Fellow}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$
$$\mathcal{R}_2 = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Prof}(\mathsf{tom}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$$

obtaining by removing one of $\mathsf{Fellow}(\mathsf{tom})$ and $\mathsf{Prof}(\mathsf{tom})$. By computing the answers to q_1 for each of the two repairs, one can show that anna and tom are both answers to q_1 under AR semantics, while cs101 is not an AR-answer.

Unfortunately, query answering under AR semantics is known to be intractable (more precisely: coNP-hard in the size of the data) even for lightweight ontology languages like DL-Lite. In fact, a single class disjointness axiom suffices to show intractability [Bienvenu, 2012], so there is no hope of regaining tractability by restricting the ontology language.

Coping with Intractability through Approximation

To cope with the intractability of the AR semantics, one can turn to approximations. A natural over-approximation of the AR semantics is given by the *brave semantics* [Bienvenu and Rosati, 2013], which returns all query answers that can be obtained from at least one repair (i.e., they are supported by some internally consistent set of facts). The more cautious *IAR semantics* [Lembo *et al.*, 2015], which queries the intersection of all repairs, provides a natural under-approximation. Both semantics have appealing computational properties: for most DL-Lite dialects, query answering using these semantics is tractable in data complexity and can be implemented by first-order query rewriting.

Example 5. The intersection of the repairs $\mathcal{R}_1$ and $\mathcal{R}_2$ yields $\mathcal{R}_\cap = \{\mathsf{Prof}(\mathsf{anna}), \mathsf{Teaches}(\mathsf{tom}, \mathsf{cs101})\}$. As $\mathsf{Prof}(\mathsf{anna})$ belongs to $\mathcal{R}_\cap$, anna is an answer to q_1 under IAR semantics, but tom is not an IAR-answer, as we cannot derive $\mathsf{Faculty}(\mathsf{tom})$ from the facts in $\mathcal{R}_\cap$. For the query $q_2(x) = \mathsf{Prof}(x)$, both anna and tom are brave answers, while only anna is an answer under AR and IAR semantics.

The CQAPri system [Bienvenu *et al.*, 2014a], which is the first to implement the AR semantics for DL-Lite$_\mathcal{R}$ ontologies, adopts a hybrid approach in which the brave and IAR semantics serve to efficiently identify a large portion of the (non)answers to a query under AR semantics, and then a SAT solver is used to decide the status of the remaining candidate answers. This approach appears quite promising, as the tractable approximations do most of the work, and the generated SAT instances are rather small and easy to solve.

To obtain more fine-grained approximations, [Bienvenu and Rosati, 2013] introduced two new parameterized families of inconsistency-tolerant semantics, called k-defeater and k-support semantics, that approximate the AR semantics from above and from below, respectively, and converge to the AR semantics in the limit. These new semantics appear quite promising, as they generalize the IAR and brave semantics, while retaining the same desirable computational properties. It will be interesting to see whether these new semantics can be profitably exploited to further reduce the number of calls to SAT solvers needed to identifying AR query answers.

At present, most of the practical work on inconsistency-tolerant OMQA has focused on DLs of the DL-Lite family, so an important challenge for future work is to design efficient methods for other popular ontology languages.

User-in-the-Loop: Explanation and Interaction

The need to equip reasoning systems with explanation services is widely acknowledged, and such facilities are all the more essential when using inconsistency-tolerant semantics. Indeed, the brave, AR, and IAR semantics allow one to classify query answers into three categories of increasing reliability, and a user may naturally wonder why a given tuple was assigned to, or excluded from, one of these categories. This problem was recently tackled by [Bienvenu *et al.*, 2016a], who devised a framework for explaining query (non-)answers in this setting and explored the computational properties of explanations when the ontology is given in DL-Lite$_\mathcal{R}$. While many of the explanation tasks proved to be intractable, they can nonetheless be solved quickly by making use of the facilities of modern SAT solvers.

In addition to helping users understand query results, an OMQA system should also provide users with a way of giving feedback on missing or erroneous results, thereby involving them in the effort to improve data quality. In [Bienvenu *et al.*, 2016b], a formal framework for query-driven repairing is proposed, in which the aim is to find a set of elementary data modifications (deletions and additions), called a repair plan, that addresses as many of the defects as possible, subject to the condition that all changes must be validated by the user. Different notions of optimality are introduced to define what it means for a repair plan to the best possible, and interactive algorithms are proposed for computing optimal repair plans.

4 Future of OMQA Research: An Invitation

Over the past decade, OMQA has grown into a very active area of research, bringing together researchers from knowledge representation, databases, and the Semantic Web. Reasoning techniques are becoming increasingly mature, allowing OMQA techniques to be experimented in real-world applications. For example, in the EU-funded Optique project, industrial partners Statoil and Siemens are adopting OMQA to make it possible for end users to formulate their queries over multiple complex data sources. Beyond their use in next-generation enterprise information systems, OMQA also holds much promise in the areas of medicine and the life sciences, where significant energy has already been spent in developing high-quality ontologies, the large-scale medical ontology SNOMED being the most prominent example.

Despite significant recent progress, there still remains much to be done to ensure the widespread adoption of OMQA in practice, and researchers from other AI areas have a lot to contribute. For instance, natural language processing methods could help make OMQA systems more accessible to inexperienced users, allowing them to interact with the system using natural language (both for posing queries and when trying to understand or debug the output). Machine learning techniques could be used to support the semi-automatic construction of both ontologies and queries, or used in conjunction with OMQA to support hybrid exploration of data. Combining the strengths of different AI subcommunities will allow us to get even more from data.

To learn more Readers can consult [Bienvenu and Ortiz, 2015] for a detailed introduction to OMQA and an overview of recent research directions. The tutorial [Kontchakov *et al.*, 2013] focuses on DL-Lite / OWL 2 QL ontologies and provide more details on the use of database systems, whereas [Ortiz and Šimkus, 2012] provides a detailed treatment of OMQA with more expressive DLs.

Acknowledgements

The author is supported by contract ANR-12-JS02-007-01.

References

[Bienvenu and Ortiz, 2015] Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Reasoning Web*, pages 218–307, 2015.

[Bienvenu and Rosati, 2013] Meghyn Bienvenu and Riccardo Rosati. Tractable approximations of consistent query answering for robust ontology-based data access. In *Proc. of IJCAI*, 2013.

[Bienvenu *et al.*, 2013a] Meghyn Bienvenu, Carsten Lutz, and Frank Wolter. First order-rewritability of atomic queries in Horn description logics. In *Proc. of IJCAI*, pages 754–760, 2013.

[Bienvenu *et al.*, 2013b] Meghyn Bienvenu, Magdalena Ortiz, Mantas Simkus, and Guohui Xiao. Tractable queries for lightweight description logics. In *Proc. of IJCAI*, 2013.

[Bienvenu *et al.*, 2014a] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Querying inconsistent description logic knowledge bases under preferred repair semantics. In *Proc. of AAAI*, 2014.

[Bienvenu *et al.*, 2014b] Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. Ontology-based data access: a study through Disjunctive Datalog, CSP, and MMSNP. *ACM Trans. Database Syst. (TODS)*, 39, 2014.

[Bienvenu *et al.*, 2015] Meghyn Bienvenu, Stanislav Kikot, and Vladimir V. Podolskii. Tree-like queries in OWL 2 QL: Succinctness and complexity results. In *Proc. of LICS*, 2015.

[Bienvenu *et al.*, 2016a] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Explaining inconsistency-tolerant query answering over description logic knowledge bases. In *Proc. of AAAI*, 2016.

[Bienvenu *et al.*, 2016b] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. Query-driven repairing of inconsistent DL-Lite knowledge bases. In *Proc. of IJCAI*, 2016.

[Bienvenu *et al.*, 2016c] Meghyn Bienvenu, Peter Hansen, Carsten Lutz, and Frank Wolter. First order-rewritability of conjunctive queries in Horn description logics. In *Proc. of IJCAI*, 2016.

[Bienvenu *et al.*, 2016d] Meghyn Bienvenu, Roman Kontchakov, Stanislav Kikot, Vladimir Podolskii, and Michael Zakharyaschev. Ontology-mediated queries: Combined complexity and succinctness of rewritings via circuit complexity. Submitted, available on the author's website, 2016.

[Bienvenu *et al.*, 2016e] Meghyn Bienvenu, Roman Kontchakov, Stanislav Kikot, Vladimir Podolskii, and Michael Zakharyaschev. Theoretically optimal datalog rewritings for OWL 2 QL ontology-mediated queries. In *Proc. of DL*, 2016.

[Bienvenu, 2012] Meghyn Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proc. of AAAI*, 2012.

[Calvanese *et al.*, 2007] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: the *DL-Lite* family. *J. of Automated Reasoning (JAR)*, 39(3):385–429, 2007.

[Hansen *et al.*, 2015] Peter Hansen, Carsten Lutz, Inanç Seylan, and Frank Wolter. Efficient query rewriting in the description logic EL and beyond. In *Proc. of IJCAI*, 2015.

[Kikot *et al.*, 2012] Stanislav Kikot, Roman Kontchakov, Vladimir Podolskii, and Michael Zakharyaschev. Exponential lower bounds and separation for query rewriting. In *Proc. of ICALP*, 2012.

[Kikot *et al.*, 2014] Stanislav Kikot, Roman Kontchakov, Vladimir Podolskii, and Michael Zakharyaschev. On the succinctness of query rewriting over shallow ontologies. In *Proc. of LICS*, 2014.

[Kontchakov *et al.*, 2013] Roman Kontchakov, Mariano Rodriguez-Muro, and Michael Zakharyaschev. Ontology-based data access with databases: A short course. In *Reasoning Web*, pages 194–229, 2013.

[Lembo and Ruzzi, 2007] Domenico Lembo and Marco Ruzzi. Consistent query answering over description logic ontologies. In *Proc. of RR*, 2007.

[Lembo *et al.*, 2015] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant query answering in ontology-based data access. *J. Web Semantics (JWS)*, 33:3–29, 2015.

[Ortiz and Šimkus, 2012] Magdalena Ortiz and Mantas Šimkus. Reasoning and query answering in description logics. In *Reasoning Web*, pages 1–53, 2012.