On the Relationship Between Ontology-mediated Queries and Non-monotonic Datalog

Shqiponja Ahmetaj, Magdalena Ortiz, and Mantas Šimkus

Institute of Information Systems, TU Vienna, Austria

In *ontology-mediated queries* (OMQs), a database query expression (which may be, for example, a relational algebra expression written as a first-order logic formula, or a conjunctive query), is coupled with an *ontology* that provides domain knowledge. The ontology can extend the vocabulary used for querying, and allows to leverage the domain knowledge to obtain more complete answers from incomplete data.

For example, consider the query $q(x) = \mathsf{teachingPers}(x)$ that aims at retrieving all personal that teaches in a university. Suppose that the university database has the following relevant tables: (1) a unary $\mathsf{professor}$ relation containing professors, (2) a unary $\mathsf{lecturer}$ relation containing lecturers, (3) a binary $\mathsf{teaches}$ relation containing tuples of the form (t, c) where t is the person teaching course c; the values of t in this table may be, for instance, external lecturers that are not in the two tables $\mathsf{professor}$ or $\mathsf{lecturer}$. The query q would not retrieve any answers when evaluated directly on the database (in fact, the $\mathsf{teachingPers}$ relation is not even present in the database). However, it would retrieve the expected answers if coupled with the following ontology

$$\mathsf{professor} \sqsubseteq \mathsf{teachingPers} \qquad \mathsf{lecturer} \sqsubseteq \mathsf{teachingPers} \qquad \exists \mathsf{teaches} \sqsubseteq \mathsf{teachingPers}$$

that says that professors and lecturers are teaching personal, and so are all the values occurring in the first column of the $\mathsf{teaches}$ relation.

OMQs are receiving much attention in the database and knowledge representation research communities, particularly when the ontological knowledge is expressed in Description Logics (DLs) as in the example above (in this case, the ontology is often called a *TBox*), or in rule-based formalisms like *existential rules* and DATALOG± (see, e.g., [4, 3, 11] and their references).

While OMQs are powerful query languages with obvious attractive features, their exact relationship with more traditional query languages is not well understood. In particular, we are interested in the *relative expressiveness* of OMQs compared to query languages like first-order queries, or plain DATALOG and its variations. More precisely, the following problem is of great importance: given an OMQ Q (specified by a database query expression, and a DL ontology or *TBox*), obtain a DATALOG query Q'—in a suitable variation of DATALOG—such that, for any set of facts $\mathcal{A}$ (a.k.a. an *ABox* in DL jargon), the certain answer to Q over $\mathcal{A}$ coincides with the certain answer to Q' over $\mathcal{A}$.

The existence of such a Q' and its size are crucial for understanding the expressive power and succinctness of different families of OMQs. However, they are also very relevant in practice, since they allow to reuse existing database technologies to support OMQ answering. For example, the research into OMQs that can be rewritten into *first-order (FO) queries* has produced the successful *DL-Lite* family [5]. The succinctness of FO-rewritings for DL-Lite, and for families of existential rules that are FO-rewritable, has been extensively studied [9, 12], and for cases where (succinct) FO-rewritings do not exists, some authors have considered rewritings that are not *data-independent* [16, 10]. Many DLs are not FO-rewritable, but can be rewritten into monotonic DATALOG queries, leading to implemented systems, e.g., [21, 7, 23]. The pioneering work in [13] showed that instance queries in an expressive extension of $\mathcal{ALC}$ can be rewritten into a program in disjunctive DATALOG, using a constant number of variables per rule, but exponentially many rules. The first translation from *conjunctive queries (CQs)* in expressive DLs $\mathcal{SH}$ and $\mathcal{SHQ}$ to programs in disjunctive DATALOG was introduced in [6], but the program may contain double exponentially many predicates and is not fully data-independent. For $\mathcal{ALC}$ and for union of CQs, the existence of data-independent exponential rewritings into disjunctive DATALOG was shown recently [4], and for restricted fragments of $\mathcal{SHI}$ and classes of CQs translations to DATALOG were investigated in [14, 15]. A polynomial time DATALOG translation of instance queries was proposed in [20], but for a so-called *Horn-DL* that lacks disjunction. Until very recently, this was the only polynomial rewriting for a DL that is not FO-rewritable.

Combining complete and incomplete information in OMQs

Ontologies in all of the languages mentioned above can be seen as first-order theories, and as such, they adopt an *open-world semantics*. In particular, the certain answers to an OMQ over an ABox $\mathcal{A}$ are computed by (i) evaluating the input database query expression over all possible models of the input ontology and $\mathcal{A}$, and (ii) computing the intersection of all obtained results. Such an *open-world* nature of OMQ languages makes them suitable for handling incomplete knowledge. However, it has been acknowledged recently that viewing *all* data as incomplete is too restrictive as the evaluation of an OMQ may, counterintuitively, result in too few certain answers. For this reason, *closed predicates* have been advocated as a powerful tool to combine complete and incomplete knowledge, by explicitly specifying predicates assumed complete, thus given a *closed-world* semantics [8, 17]. For example, take the following TBox $\mathcal{T}$

$$\begin{aligned}\mathsf{BScStud} &\sqsubseteq \mathsf{Student}\\ \mathsf{Student} &\sqsubseteq \exists \mathsf{attends}.\mathsf{Course}\\ \mathsf{BScStud} &\sqsubseteq \forall \mathsf{attends}.\neg \mathsf{GradCourse}\end{aligned}$$

and the following ABox $\mathcal{A}$:

$$\begin{array}{ll}\mathsf{Course}(c_1) & \qquad\mathsf{BScStud}(a)\\ \mathsf{Course}(c_2) & \qquad\mathsf{GradCourse}(c_2)\end{array}$$

Then (a, c_1) is not a certain answer to the *instance query* $q(x, y) = \mathsf{attends}(x, y)$ mediated by $\mathcal{T}$, but if c_1 and c_2 are known to be the *only* courses, then (a, c_1)

should become a certain answer. This can be achieved by declaring Course a *closed predicate*.

There are only a few relative expressiveness results for OMQs with closed predicates. FO-rewritability for the core fragment of DL-Lite was presented in [18], and a rewriting algorithm for queries that satisfy some strong *definability* criteria we given in [22]. Other works on OMQs with closed predicate have focused on the complexity of their evaluation, e.g., [19, 17, 8]. The latter two have shown coNP-hardness in data complexity for many lightweight DLs, barring the existence of FO-rewritings. Recently, we have studied the class $\mathcal{Q}$ of OMQs of the form $(\mathcal{T}, \Sigma, q)$, where q is an instance query and $\mathcal{T}$ is a TBox in the very expressive DL $\mathcal{ALCHIO}$ with closed predicates Σ. We observed that such queries are *non-monotonic*. Indeed, if we take $\Sigma = \{\mathsf{Course}\}$ as the set of closed predicates in the above example, then (a, c_1) is a certain answer to $(\mathcal{T}, \Sigma, q)$ over $\mathcal{A}$, but it is not a certain answer over the extended set of facts $\mathcal{A}' = \mathcal{A} \cup \{\mathsf{Course}(c_3)\}$. For this reason, these queries cannot be rewritten into monotonic variants of DATALOG, like positive DATALOG (with or without disjunction). The main contribution of [1] was a polynomial time translation of queries in $\mathcal{Q}$ into *disjunctive* DATALOG *extended with negation under the stable model semantics*. Our translation is modular: if no closed predicates are present—in the case of classical instance queries in $\mathcal{ALCHIO}$—our translation yields a positive disjunctive DATALOG program of polynomial size. A simplified version of this translation for $\mathcal{ALCHI}$ can be found in [2]. To our knowledge, this is the first polynomial time translation of an expressive (non-Horn) DL into disjunctive DATALOG.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF) projects P25207, T515 and W1255.

References

1. Shqiponja Ahmetaj, Magdalena Ortiz, and Mantas Šimkus. Polynomial datalog rewritings for expressive description logics with closed predicates. In *Proc. of IJCAI 2016*, pages 878–885. IJCAI/AAAI Press, 2016.
2. Shqiponja Ahmetaj, Magdalena Ortiz, and Mantas Šimkus. Polynomial disjunctive datalog rewritings of instance queries in expressive description logics. In *Proc. of DL 2016*, 2016.
3. Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Reasoning Web*, volume 9203 of *Lecture Notes in Computer Science*, pages 218–307. Springer, 2015.
4. 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.*, 39(4):33:1–33:44, 2014.
5. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

6. Thomas Eiter, Magdalena Ortiz, and Mantas Šimkus. Conjunctive query answering in the description logic SH using knots. *J. Comput. Syst. Sci.*, 78(1):47–85, 2012.
7. Thomas Eiter, Magdalena Ortiz, Mantas Šimkus, Trung-Kien Tran, and Guohui Xiao. Query rewriting for Horn-SHIQ plus rules. In *Proc. of AAAI 2012*. AAAI Press, 2012.
8. Enrico Franconi, Yazmin Angélica Ibáñez-García, and Inanç Seylan. Query answering with DBoxes is hard. *Electr. Notes Theor. Comput. Sci.*, 278:71–84, 2011.
9. Georg Gottlob, Stanislav Kikot, Roman Kontchakov, Vladimir V. Podolskii, Thomas Schwentick, and Michael Zakharyaschev. The price of query rewriting in ontology-based data access. *Artif. Intell.*, 213:42–59, 2014.
10. Georg Gottlob, Marco Manna, and Andreas Pieris. Polynomial combined rewritings for existential rules. In *Proc. of KR 2014*. AAAI Press, 2014.
11. Georg Gottlob, Marco Manna, and Andreas Pieris. Polynomial rewritings for linear existential rules. In *Proc. of IJCAI 2015*. AAAI Press, 2015.
12. Georg Gottlob and Thomas Schwentick. Rewriting ontological queries into small nonrecursive datalog programs. In *Proc. of KR 2012*. AAAI Press, 2012.
13. Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Reasoning in description logics by a reduction to disjunctive datalog. *J. Autom. Reasoning*, 39(3):351–384, 2007.
14. Mark Kaminski, Yavor Nenov, and Bernardo Cuenca Grau. Computing datalog rewritings for disjunctive datalog programs and description logic ontologies. In *Proc. of RR 2014*, pages 76–91, 2014.
15. Mark Kaminski, Yavor Nenov, and Bernardo Cuenca Grau. Datalog rewritability of disjunctive datalog programs and its applications to ontology reasoning. In *Proc. of AAAI 2014*, pages 1077–1083, 2014.
16. Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to ontology-based data access. In *Proc. of IJCAI 2011*. IJCAI/AAAI, 2011.
17. Carsten Lutz, Inanç Seylan, and Frank Wolter. Ontology-based data access with closed predicates is inherently intractable(sometimes). In *Proc. of IJCAI 2013*. IJCAI/AAAI, 2013.
18. Carsten Lutz, Inanç Seylan, and Frank Wolter. Ontology-mediated queries with closed predicates. In *Proc. of IJCAI 2015*. IJCAI/AAAI, 2015.
19. Nhung Ngo, Magdalena Ortiz, and Mantas Šimkus. The combined complexity of reasoning with closed predicates in description logics. In *Proc. of DL 2015*. CEUR-WS.org, 2015.
20. Magdalena Ortiz, Sebastian Rudolph, and Mantas Šimkus. Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In *Proc. of KR 2010*. AAAI Press, 2010.
21. Héctor Pérez-Urbina, Boris Motik, and Ian Horrocks. Tractable query answering and rewriting under description logic constraints. *J. Applied Logic*, 8(2):186–209, 2010.
22. Inanç Seylan, Enrico Franconi, and Jos de Bruijn. Effective query rewriting with ontologies over DBoxes. In *Proc. of IJCAI 2009*, 2009.
23. Despoina Trivela, Giorgos Stoilos, Alexandros Chortaras, and Giorgos B. Stamou. Optimising resolution-based rewriting algorithms for OWL ontologies. *J. Web Sem.*, 33:30–49, 2015.