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 \( q(x) = \text{teachingPers}(x) \) that aims at retrieving all personal that teaches in a university. Suppose that the university database has the following relevant tables: (1) a unary professor relation containing professors, (2) a unary lecturer relation containing lecturers, (3) a binary 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 professor or lecturer. The query \(q\) would not retrieve any answers when evaluated directly on the database (in fact, the teachingPers relation is not even present in the database). However, it would retrieve the expected answers if coupled with the following ontology

\[
\text{professor} \sqsubseteq \text{teachingPers} \quad \text{lecturer} \sqsubseteq \text{teachingPers} \quad \exists\text{teaches} \sqsubseteq \text{teachingPers}
\]

that says that professors and lecturers are teaching personal, and so are all the values occurring in the first column of the 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 \(A\) (a.k.a. an ABox in DL jargon), the certain answer to \(Q\) over \(A\) coincides with the certain answer to \(Q'\) over \(A\).

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 exist, 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 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 $SH$ and $SHQ$ 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 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 $SHI$ 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 $A$ are computed by (i) evaluating the input database query expression over all possible models of the input ontology and $A$, 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 $T$

\[
\begin{align*}
\text{BScStud} &\subseteq \text{Student} \\
\text{Student} &\supseteq \exists \text{attends}.\text{Course} \\
\text{BScStud} &\subseteq \forall \text{attends}.\neg \text{GradCourse}
\end{align*}
\]

and the following ABox $A$:

\[
\begin{align*}
\text{Course}(c_1) &\quad \text{BScStud}(a) \\
\text{Course}(c_2) &\quad \text{GradCourse}(c_2)
\end{align*}
\]

Then $(a, c_1)$ is not a certain answer to the instance query $q(x, y) = \text{attends}(x, y)$ mediated by $T$, 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 Q of OMQs of the form \((T, \Sigma, q)\), where \(q\) is an instance query and \(T\) is a TBox in the very expressive DL \(\text{ALCHIO}\) with closed predicates \(\Sigma\). We observed that such queries are non-monotonic. Indeed, if we take \(\Sigma = \{\text{Course}\}\) as the set of closed predicates in the above example, then \((a,c_1)\) is a certain answer to \((T, \Sigma, q)\) over \(A\), but it is not a certain answer over the extended set of facts \(A' = A \cup \{\text{Course}(c_3)\}\). 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 \(Q\) 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 \(\text{ALCHIO}\)—our translation yields a positive disjunctive Datalog program of polynomial size. A simplified version of this translation for \(\text{ALCHI}\) 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