Relaxing and Restraining Queries for OBDA — Extended Abstract

Medina Andreșel, Yazmín Ibáñez-García, Magdalena Ortiz, Mantas Šimkus
{andresel,ibanez,ortiz}@kr.tuwien.ac.at | simkus@dbai.tuwien.ac.at
Faculty of Informatics, TU Wien, Austria

Abstract

We investigate query reformulation rules in OBDA to obtain either more or less answers. We extend DL-Lite with complex role inclusions and define rules that produce query relaxations/restrictions over any dataset. We also introduce a set of data-driven rules to get more fine-grained reformulations.

In Ontology-based data access (OBDA) an ontology provides a conceptual view of a collection of data sources, and describes knowledge about the domain of interest at a high level of abstraction. Thus users can formulate queries over data sources using a familiar vocabulary provided by the ontology, while the represented knowledge can be leveraged to retrieve more complete answers. For example, consider the following dataset about cultural events and their locations,

\[ A_c = \{\text{Concert}(c_1), \text{Venue}(\text{StateOpera}), \text{Exhibition}(e_1), \text{City}(\text{Vienna}), \text{CulturEvent}(\text{ev}_1), \text{Country}(\text{Austria}), \text{occursIn}(c_1, \text{StateOpera}), \text{occursIn}(\text{ev}_1, \text{Vienna}), \text{locIn}(\text{StateOpera}, \text{Vienna}), \text{locIn}(\text{Vienna, Austria})\}, \]

and the ontology \( T_c \) below, which captures, among other things, the knowledge that both concerts and exhibitions are cultural events.

\[
\begin{align*}
\text{Country} & \sqsubseteq \text{Location} & \text{Exhibition} & \sqsubseteq \text{CulturEvent} \\
\text{Venue} & \sqsubseteq \text{Location} & \text{Theater} & \sqsubseteq \text{Venue} \\
\text{City} & \sqsubseteq \text{Location} & \text{Museum} & \sqsubseteq \text{Venue} \\
\text{Concert} & \sqsubseteq \text{CulturEvent} & \text{CulturEvent} & \sqsubseteq \text{Event} \\
\exists \text{locIn} & \sqsubseteq \text{Location} & \exists \text{locIn}^- & \sqsubseteq \text{Location} \\
\exists \text{occursIn} & \sqsubseteq \text{Event} & \exists \text{occursIn}^+ & \sqsubseteq \text{Location}
\end{align*}
\]

Using this knowledge one can retrieve all cultural events, \( e_1, \text{ev}_1 \), and \( c_1 \), by posing the query:

\[ q_1(x) \leftarrow \text{CulturEvent}(x). \]

Description logics (DLs) of the DL-Lite family have been particularly tailored for OBDA (Calvanese et al. 2007). As a result, queries mediated by DL-Lite ontologies are first-order (FO)-rewritable. This means that evaluating a query \( q \)

\( \begin{align*}
\text{(S1)} & \text{ if } A_1 \sqsubseteq A_2 \in T \text{ and } A_2(x) \in q, \text{ then } \theta = [A_2(x)/A_1(x)]; \\
\text{(S2)} & \text{ if } A \sqsubseteq \exists r \in T \text{ and } r(x, y) \in q, \text{ and } y \text{ is a non-answer variable occurring only once in } q, \text{ then } \theta = [r(x, y)/A(x)]; \\
\text{(S3)} & \text{ if } \exists r \sqsubseteq A \in T \text{ and } A(x) \in q, \text{ then } \theta = [A(x)/r(x, z^\theta)]; \\
\text{(S4)} & \text{ if } r \sqsubseteq s \in T \text{ and } s(x, y) \in q, \text{ then } \theta = [s(x, y)/r(x, y)]; \\
\text{(S5)} & \text{ if } r \sqsubseteq s^- \in T \text{ and } s(x, y) \in q, \text{ then } \theta = [s(x, y)/r(y, x)]; \\
\text{(S6)} & \text{ if } t \cdot s \sqsubseteq r \in T \text{ and } r(x, y) \in q, \text{ then } \theta = [r(x, y)/\{t(x, z^\theta), s(z^\theta, y)}]].
\end{align*} \]

where \( z^\theta \) is a fresh variable.

Table 1: Rewriting rules for DL-Lite

TBox \((T, A)\) can be reduced to evaluate a query \( q_T \) (incorporating knowledge from \( T \)) over \( A \) alone, which amounts to query evaluation in relational databases. In our example, a rewriting of \( q_1 \) is

\( q_T(x) \leftarrow \text{CulturEvent}(x) \lor \text{Exhibition}(x) \lor \text{Concert}(x) \)

In this paper we investigate the use of rewritings in OBDA for relaxing and constraining queries to, respectively, retrieve more or less answers\(^1\). A key observation in our approach is that query restrictions can be obtained using existing rewriting rules and that ‘counterparts’ of these rules can be defined to produce relaxations. In our example, the query \( q_{c_1}(x) \leftarrow \text{Concert}(x) \) that restricts \( q_1 \) occurs as a disjunct in \( q_T \).

Using the perfect reformulation for DL-Lite proposed by (Calvanese et al. 2007), a rewriting \( q' \) of \( q \) is obtained by applying an atom substitution \( \theta \) to \( q \) as described in rules (S1) – (S5) in Table 1. A way to define a counterpart e.g., for (S3) to obtain a relaxation of a given query \( q \) w.r.t. a TBox \( T \) is the rule (G2): replace \( A(x) \in q \) by \( r(x, y) \) if \( A \sqsubseteq \exists r \in T \), with \( y \) a fresh variable. We also define counterparts of rules S1–S5 to obtain query relaxations (see G1–G5 in the full version of this paper).

Notably, there are intuitive answers and reformulations that cannot be produced with the standard DL-Lite rewriting rules and their counterparts. For example, consider a query retrieving concerts occurring in Vienna:

\[ q_2(x) \leftarrow \text{Concert}(x), \text{occursIn}(x, y), y = \text{Vienna}. \]

\( ^1\)This research was funded by FWF Projects P30360 and W1255-N23

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
There are no answers to \( q_2 \) when evaluated over \((T_e, A_e)\), although \( c_i \) may be considered an answer to \( q_2 \) according to the intuition that if an event occurs in a venue located in a city, then it occurs in that city. In order to enable this kind of reformulations, we extend the expressive power of DL-Lite with complex role inclusions (CRIs). In our example, we could add the following:

\[
\text{occursIn} \cdot \text{locIn} \sqsubseteq \text{occursIn}
\]

(1)
to capture the intuition above. We call the extension of DL-Lite with CRIs DL-Lite\(^{\text{HR}}\), and propose reformulation rules operating not only along the subclass (subrole) relation, but also along CRIs (see (S6) in Table 1). We can now use (1) and (S6) to restrain the query

\[
q_3(x) \leftarrow \text{Concert}(x), \text{occursIn}(x,y), \text{City}(y),
\]

from all concerts occurring in a city, to only those for which a more specific location within a city is known:

\[
q_3'(x) \leftarrow \text{Concert}(x), \text{occursIn}(x,z), \text{locIn}(z,y), \text{City}(y)
\]

The following rule is the dual of (S6), namely (G6) replaces \( \{r(x,y), s(y,z)\} \sqsubseteq q \) by \( r(x,z) \), if \( y \) is a non-answer variable that does not occur elsewhere in \( q \) and \( r \cdot s \sqsubseteq r \in T \). We show that our ontology-driven rules restrain (or relax) a certain answers of \( q \) w.r.t. \((T, A)\) are necessarily contained in (or contain) the certain answers of \( q \).

It is well-known that unrestricted usage of CRI can easily lead to undecidability (Horrocks and Sattler 2004). Even when imposing syntactic restrictions to CRIs to regain decidability, such as regularity (Kazakov 2010), query answering is not FO-rewritable in DL-Lite\(^{\text{HR}}\) (Artale et al. 2009). In the full version of this paper, we propose suitable restrictions to CRIs to regain FO-rewritability.

We also consider data-driven reformulations, which unlike the ontology-based ones, are specific to a dataset at hand. For example, the query \( q_2 \) asking for concerts occurring in Vienna could be restrained to concerts in the State Opera in Vienna, or relaxed to all concerts in Austria. These reformulations cannot be done on the basis of the TBox alone since they depend on the specific dataset. In our example, they are based on the assertions \( \text{locIn(Vienna, Austria)} \) and \( \text{locIn(StateOpera, Vienna)} \).

Our data-driven rules use assertions as follows for \( K = (T, A) \): (SD1) if \( A(x) \in q \) and \( K \models A(a) \), then we can restrain \( q \) by adding the atom \( x = a \); and (SD2) if \( r(x,y) \in q \) and \( K \models r(a,b) \), then we can add \( x = a \) or \( y = b \). For relaxing, (GD1) replaces \( x = a \) with \( A(x) \), if \( K \models A(a) \); and (GD2) replaces \( x = a \in q \) by the atoms \( r(x,y), y = b \), if \( K \models r(a,b) \). For example, using (GD2) and \( \text{locIn(Vienna, Austria)} \in A_e \), we can relax the query from the concerts in Vienna,

\[
q(x) \leftarrow \text{Concert}(x), \text{occursIn}(x,y), y = \text{Vienna}
\]
to concerts that occur in a location in Austria:

\[
q'(x) \leftarrow \text{Concert}(x), \text{occursIn}(x,y), \text{locIn}(y,z), z = \text{Austria}
\]

Besides concepts and role assertions, we consider as well dependencies that are not necessarily implied by the ontology, but that can be guaranteed to hold in the current dataset. A quick inspection at \( A_e \) reveals that every existing venue is located in a city. We could use this knowledge to restrain the query

\[
q(x) \leftarrow \text{Event}(x), \text{occursIn}(x,y), \text{locIn}(y,z), \text{City}(z) \quad \text{into} \quad q'(x) \leftarrow \text{Event}(x), \text{occursIn}(x,y), \text{Venue}(y)
\]

Indeed, such a reformulation could be done based on the ontology alone, provided that the axiom \( \text{Venue} \sqsubseteq \text{City} \) is present. However, we may not have such an axiom, and it may not be possible or desirable to add it. Then, for a given \( K = (T, A) \), we define rules using query containment tests, \( q_1(x) \sqsubseteq_K q_2(x) \), where \( q_1, q_2 \) are queries with at most two atoms and two variables. We remark these tests are not expensive in the extension of DL-Lite with CRI that is FO-rewritable. These rules are very similar to the ones using only the ontology, except they allow to replace \( B(x) \) by \( A(x) \) for restraining, not only when \( A \sqsubseteq B \) in \( T \), but also when the weaker condition \( A(x) \sqsubseteq_K B(x) \) holds. Note that these replacements are also allowed for some more complex pairs of atoms. For instance, if \( q^*(x) \leftarrow r(x,y), B(y) \) and \( A(x) \sqsubseteq_K q^*(x) \), rule (GD4) will replace \( A(x) \in q \) with \( r(x,y), B(y) \) producing a relaxation. Consider for example the query

\[
q(x) \leftarrow \text{Event}(x), \text{occursIn}(x,y), \text{City}(y)
\]

if \( \text{City}(x) \sqsubseteq_K \exists \text{locIn.City}(x) \), then using rule (GD4) we obtain:

\[
q'(x) \leftarrow \text{Event}(x), \text{occursIn}(x,y), \text{locIn}(y,z), \text{Country}(z)
\]

Now, we can actually apply (G6) obtaining the query

\[
q''(x) \leftarrow \text{Event}(x), \text{occursIn}(x,z), \text{Country}(z).
\]

Thus, our data-driven rules are useful for query reformulation not only on their own, but also because they may trigger other relevant reformulations that were not obtainable otherwise. For the data-driven rules the containment only holds when evaluated over \((T, A) \), but not for an arbitrary \( A \).

The proposed reformulations can aid users to explore heterogeneous, unstructured and incomplete datasets in the same spirit as online analytical processing (OLAP) supports the exploration of structured data from multiple perspectives and at various granularity levels (Codd, Codd, and Salley 1993). For that purpose, our extension of DL-Lite takes into account dimensional knowledge, analogous to the so-called multidimensional data model (Hurtado and Mendelzon 2002), and our reformulation rules are designed in such a way that they emulate so-called ‘rolling-up’ and ‘drilling-down’ operations along dimensions.

References


Kazakov, Y. 2010. An extension of complex role inclusion axioms in the description logic SROIQ. In Proc. of IJCAR.