Object-Relational Queries over $\mathcal{CFDI}_{nc}^{\forall-}$ Knowledge Bases: OBDA for the SQL-Literate (extended abstract)[1]

Jason St. Jacques, David Toman and Grant Weddell
Cheriton School of Computer Science
University of Waterloo, Canada
jw.stjacques@gmail.com,{david,gweddell}@uwaterloo.ca

Overview and Main Results. *Ontology based data access* (OBDA) is concerned with computing query answers over (possibly incomplete) data sources for which *background knowledge* about the data, commonly captured in an ontology, is available. The background knowledge enhances the understanding of the data source, provides additional query answers that may not be explicit in the data itself, and can also simplify query formulation.

To address scalability issues relating to the volume of data, many current approaches to OBDA focus on *conjunctive queries* (CQs) and ontologies based on DL dialects for which CQ answering is in AC_0/PTIME with respect to data complexity. Moreover, to leverage advances in query processing in relational systems, approaches in which query answering can be reduced to SQL query evaluation over a relational encoding of the data are commonly sought.

There are two lines of investigation in this area that have received considerable attention: (i) the *perfect rewriting*-based approaches in which the given CQ is rewritten with the help of the ontological knowledge (typically formulated in one of the DL-Lite family of logics) in such a way that the resulting query can be executed over the plain data yielding the desired answers [3], and (ii) the *combined* approaches in which the data is completed using the ontological knowledge (formulated in DL-Lite or $\mathcal{EL}$ logics) in such a way that the original query (modulo ontology-independent filtering) can be executed over the data completion [5, 6, 8, 9].

In this paper, we outline how a *combination* of query rewriting and data completion can be used to enable OBDA *directly* over a relational data source in which background knowledge is expressed in terms of $\mathcal{CFDI}_{nc}^{\forall-}$ [13], a dialect of the $\mathcal{CFD}$ family of DLs [4, 10, 12] that has PTIME complexity for many of the fundamental reasoning tasks, and that properly contains DL-Lite$_{core}^{\mathcal{F}}$. Indeed, it is worth noting that, for CQs over $\mathcal{CFDI}_{nc}^{\forall-}$ KBs, OBDA *cannot* be accomplished by either using (perfect) query rewriting alone, due to PTIME-completeness of CQ answering, nor by exclusive use of the combined approach, due to the need to realize exponentially many prototypical anonymous witnesses to represent types induced by value restrictions in a $\mathcal{CFDI}_{nc}^{\forall-}$ TBox.

We solve this problem by introducing a novel technique based on combining query rewriting with data completion. This is achieved in a three-step process by proceeding (in a purely virtual sense) through the lens of a $\mathcal{CFDI}_{nc}^{\forall-}$ ABox as follows:

- We first exhibit an ABox completion procedure for a given $\mathcal{CFDI}_{nc}^{\forall-}$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with PTIME data complexity; the completion also serves as a basis for KB consistency checking.

[1] This is an extended abstract for a paper to be presented at IJCAI 2016.

– We then define a query rewriting that produces a union of conjunctive queries Q' from a given conjunctive query Q and $\mathcal{T}$, and show that evaluating Q' as a SQL query over the above ABox completion, viewed as a relational database, computes the certain answers of Q over $\mathcal{K}$. A novel feature of this rewriting relates to the generation of CQs to account for standard functional dependencies over relational data sources, and is crucial in developing an OBDA framework that can entirely avoid any need for object/individual invention or for "named nulls."
– And finally, we show how a standard relational database schema can be naturally captured as a (fragment of a) $\mathcal{CFDI}_{nc}^{\forall -}$ TBox in a way that essentially obviates the need for additional *mappings* between data sources and virtual ABoxes that are typically utilized at this point, e.g., by [2]. We then show how rewritten queries can be executed over an underlying relational representation *without* the above-mentioned need for object invention.

Experimental results relating to the LUBM benchmark are also given that confirm the practicality of ABox completion by a direct manipulation of a relational data source. In particular, the results suggest that execution time for ABox completion in this way is comparable to the time required for raw data loading.

On the Utility of $\mathcal{CFDI}_{nc}^{\forall -}$ in OBDA over Relational Data Sources. We illustrate this with the use of a hypothetical (object) relational schema, given in Figure 1, in which single arrows denote named features, double arrows inheritance between tables, and where primary keys are underlined. We assume the schema derives from "create table" commands with primary and foreign key declarations, such as the following in the case of tables CLASS and CHAIR:

```
create table CLASS (
   dname STRING, num INT, iname STRING, room INT, time INT,
   primary key (dname, num),
   foreign key (dname) to DEPT, foreign key (iname) to PROF )
create table CHAIR (
   name STRING, dname STRING, rname STRING,
   primary key (name),
   foreign key (name) to PROF ).
```

In a $\mathcal{CFDI}_{nc}^{\forall -}$ TBox, tables such as CLASS correspond to primitive concepts, while attributes such as $dname$ and $dept$ correspond to *concrete* and *abstract* features.[2] Here are some examples of inclusion dependencies in the TBox for this schema:

1. To capture that table CLASS has attribute num and a foreign key to table DEPT:

$$\texttt{CLASS} \sqsubseteq (\forall num.\texttt{INT}) \sqcap (\forall dept.\texttt{DEPT}).$$

   Note that an (abstract) feature $dept$ is introduced to capture the foreign key, and that it is realized, in the relational schema, by a (concrete) feature $dname$ (implicit in the diagram in Figure 1).

2. To capture, respectively, the foreign and primary key declarations for table CHAIR together with a requirement that there is at least one tuple in the DEPT table that

[2] $\mathcal{CFDI}_{nc}^{\forall -}$ is a dialect of the $\mathcal{CFD}$ family of DLs, and, as such, replaces *roles* that are interpreted as binary relations with *features* that are interpreted as unary functions.

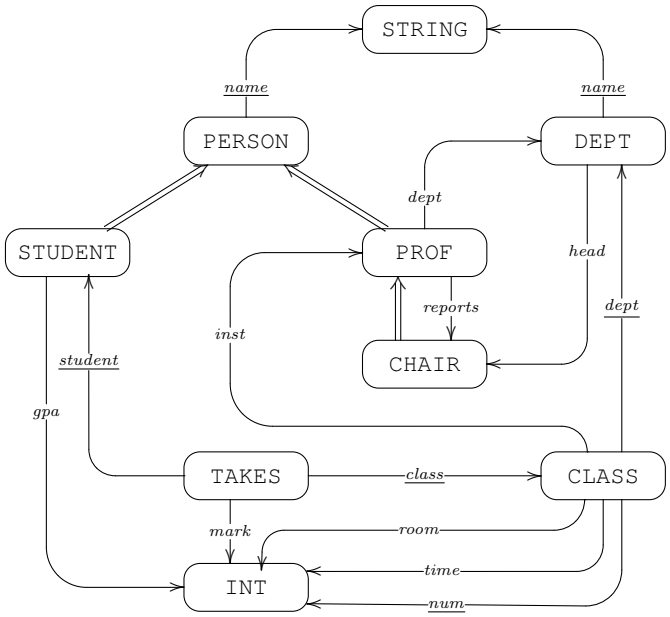

Fig. 1. A RELATIONAL SCHEMA.

refers to each tuple in CHAIR:

$$\texttt{CHAIR} \sqsubseteq \texttt{PROF} \sqcap (\texttt{CHAIR} : \mathit{name} \rightarrow \mathit{id}),$$
$$\texttt{CHAIR} \sqsubseteq \exists \mathit{head}^{-1} \text{ and } \forall \mathit{head}.\texttt{CHAIR} \sqsubseteq \texttt{DEPT}.$$

Note that we employ the PFD concept constructor available in all $\mathcal{CFD}$ dialects to capture a primary key. Its use in this case asserts that *no distinct pair of* CHAIR *objects can agree on* name*-values.*

3. To capture that *name* values in table PERSON are disjoint from *name* values in table DEPT:

$$\texttt{PERSON} \sqsubseteq \neg\texttt{DEPT} \sqcap (\texttt{DEPT} : \mathit{name} \rightarrow \mathit{id}).$$

Note again the use of the PFD concept constructor which, in this case asserts that *no combination of a* PERSON *object and* DEPT *object can have the same* name*-values.*

Future work. We briefly discuss how our adoption of $\mathcal{CFDI}_{nc}^{\forall -}$ enables further optimizations on generated SQL queries that are outlined in [4, 7, 11], in particular, that can be applied to reason about avoiding expensive *duplicate elimination*: removing `distinct` keywords, replacing `union` operations by `union all` operations, and so on.

We also outline an avenue for further work in which techniques based on so-called *referring expressions* recently proposed in [1] can be used to relax a *primary key compatibility* condition for any relational data source. The condition requires any pair of tables in the source with a simply taxonomic relationship to have identical primary key declarations, and is a common consequence of relational schemata that derive from ER modeling.

References

1. Alexander Borgida, David Toman, and Grant Weddell. On referring expressions in query answering over first order knowledge bases. In *Principles of Knowledge Representation and Reasoning*, 2016. (in press).
2. Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Davide Lanti, Martín Rezk, and Guohui Xiao. How to stay ontop of your data: Databases, ontologies and more. In *The Semantic Web: ESWC 2015 Satellite Events - ESWC 2015 Satellite Events Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*, pages 20–25, 2015.
3. 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.
4. Vitaliy L. Khizder, David Toman, and Grant Weddell. Reasoning about Duplicate Elimination with Description Logic. In *Rules and Objects in Databases (DOOD, part of CL'00)*, pages 1017–1032, 2000.
5. Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to query answering in DL-Lite. In *Principles of Knowledge Representation and Reasoning*, pages 247–257, 2010.
6. Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to ontology-based data access. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 2656–2661, 2011.
7. Huizhu Liu, David Toman, and Grant Weddell. Fine Grained Information Integration with Description Logic. In *Description Logics 2002*, pages 1–12. CEUR-WS vol.53, 2002.
8. Carsten Lutz, Inanç Seylan, David Toman, and Frank Wolter. The combined approach to OBDA: Taming role hierarchies using filters. In *International Semantic Web Conference (1)*, pages 314–330, 2013.
9. Carsten Lutz, David Toman, and Frank Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 2070–2075, 2009.
10. David Toman and Grant E. Weddell. Applications and extensions of PTIME description logics with functional constraints. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 948–954, 2009.
11. David Toman and Grant E. Weddell. *Fundamentals of Physical Design and Query Compilation*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.
12. David Toman and Grant E. Weddell. Conjunctive Query Answering in $\mathcal{CFD}_{nc}$: A PTIME Description Logic with Functional Constraints and Disjointness. In *Australasian Conference on Artificial Intelligence*, pages 350–361, 2013.
13. David Toman and Grant E. Weddell. On adding inverse features to the description logic $\mathcal{CFD}^{\forall}_{nc}$. In *PRICAI 2014: Trends in Artificial Intelligence - 13th Pacific Rim International Conference on Artificial Intelligence, Gold Coast, QLD, Australia*, pages 587–599, 2014.