Beyond OWL 2 QL in OBDA: Rewritings and Approximations

Elena Botoeva, Diego Calvanese, Valerio Santarelli, Domenico F. Savo, Alessandro Solimando, and Guohui Xiao

Ontology Based Data Access – OBDA

Logical transparency in accessing data:

- does not know where and how data is stored;
- can only see a conceptual view of data.

Ontology

- global vocabulary
- conceptual view

Mappings

- how to populate the ontology

Data Sources

- external and heterogeneous

An OBDA specification is a triple \((T, M, S)\), where:

- \(T\) is a Description Logic TBox,
- \(M\) is a set of mapping assertions \(\text{SQL}_\mathbb{A}(x) \rightarrow A(x)\) and \(\text{SQL}_\mathbb{B}(y) \rightarrow B(x, y)\), and
- \(S\) is a relational schema.

Virtual Approach to OBDA

Query Answering is done by Query Rewriting into SQL

Rewritten Query

Evaluation

- avoids data materialization
- works well for Big Data

Disadvantages:

- works only for first-order (FO) rewritable languages
- the commonly adopted one is OWL 2 QL, which is too restrictive

Beyond OWL 2 QL: Mappings to the Rescue

Problem

We want to go beyond OWL 2 QL in Virtual OBDA.

However expressive ontologies are not FO-rewritable.

Solution

Exploit the mapping component that makes use of arbitrary SQL queries.

We introduce a framework for rewriting and approximation of OBDA specifications.

Rewriting

The new specification is equivalent to the original one w.r.t. query answering (query-inseparable).

Approximation

The new specification is a sound approximation of the original one w.r.t. query answering.

\(T = \{ A \cap B \subseteq C \} \)
\(M = \{ \text{SQL}_\mathbb{A}(x) \rightarrow A(x), \text{SQL}_\mathbb{B}(y) \rightarrow B(x, y)\} \)
\(\Rightarrow T' = \{ \} \)
\(M' = \{ \text{SQL}_\mathbb{A}(x) \rightarrow A(x), \text{SQL}_\mathbb{B}(y) \rightarrow B(x, y)\} \)
\(T = \{ \exists R.A \subseteq C \} \)
\(M = \{ \text{SQL}_\mathbb{A}(x) \rightarrow A(x), \text{SQL}_\mathbb{B}(y) \rightarrow B(x, y)\} \)
\(\Rightarrow T' = \{ \} \)
\(M' = \{ \text{SQL}_\mathbb{A}(x) \rightarrow A(x), \text{SQL}_\mathbb{B}(y) \rightarrow B(x, y)\} \)
\(T = \{ \exists R.A \subseteq A \} \)
\(M = \{ \text{SQL}_\mathbb{A}(x) \rightarrow A(x), \text{SQL}_\mathbb{B}(y) \rightarrow B(x, y)\} \)
\(\Rightarrow T' = \{ \} \)
\(M' = \{ \text{SQL}_\mathbb{A}(x) \rightarrow A(x), \text{SQL}_\mathbb{B}(y) \rightarrow B(x, y)\} \)

ontoProx

We have developed an algorithm for computing approximations and implemented it in a prototype system called ontoProx.

Evaluation

We evaluated ontoProx over synthetic and real OBDA instances against

- the default ontop behavior
- local semantic approximation (LSA)
- global semantic approximation (GSA)
- clipper over materialized ABoxes.

The evaluation showed that we are able to obtain more answers using our approach (in fact, complete, whenever that could be verified by clipper).

http://ontop.inf.unibz.it
https://github.com/ontop/ontoprox
https://github.com/ghxiao/clipper

onto

Optique

unibz