DL-Lite Full: A Sub-language of OWL 2 Full for Powerful Meta-modeling

Zhenzhen Gu\textsuperscript{1} & Songmao Zhang\textsuperscript{2}
\textsuperscript{1}Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, Italy
\textsuperscript{2}Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences, Beijing, China
zhgu@unibz.it, smzhang@math.ac.cn

Background and Research Problem

Meta-modeling, usually denoting multiple uses of names like using a name as a class, property and individual simultaneously, has been studied in various description logics (DLs). However, the other kind of meta-modeling, i.e., non-standard uses of RDF/S/OWL vocabulary terms like asserting \textit{p} to be a sub-property of \texttt{rdfs\_subClassOf}, has rarely been discussed.

In order to capture the massive and varied meta-modeling described in the real-world and large-scale knowledge bases (KBs), we address the problem of extending not only multiple uses of names but also non-standard uses of \texttt{rdf\_type} in the light-weight DL DL-Lite\textsubscript{\textit{Q}}.

Main Contributions

- We provide a way of encoding multiple uses of names and non-standard uses of \texttt{rdf\_type} in DL-Lite\textsubscript{\textit{Q}} and propose a sub-language of OWL 2 Full called DL-Lite Full. For meta-knowledge accessing, meta-queries are introduced.
- For data-layer scalability, the basic idea is to extend the classical query rewriting algorithm \textit{PerfectRef} to generate new queries, but also rewritten as \textit{PerfectLit} to evaluate queries over the data-layers of KBs via meta-query rewriting and partial variable materialization.
- We prove that the considered reasoning tasks in DL-Lite Full still have AC\textsubscript{0} data complexity and PTime KB complexity.

DL-Lite Full and Meta-queries

DL-Lite Full is defined by extending DL-Lite\textsubscript{\textit{Q}} in the following two aspects:

- Do not distinguish the names for classes, roles and individuals. This means that DL-Lite Full classes, roles and individuals are defined from a same name set \textit{N}.
- \texttt{rdf\_type} can occur in the right-hand sides of inclusion axioms and be used as individuals.

By this way, multiple uses of names as well as the specifications of \texttt{rdf\_type}, such as the following axioms, can be captured by DL-Lite Full:

\[\texttt{sem:hasActor} \sqsubseteq \texttt{rdfs\_type} \texttt{person}, \texttt{sem:hasActor} \sqsubseteq \texttt{rdfs\_type} \texttt{person}\]

A DL-Lite Full KB \(K = (T, A)\) consists of a TBox \(T\) which is a set of class (role) inclusion axioms \(\sqsubseteq_{\text{rdfs\_type}}\) and an ABox \(A\) which is a set of individual assertions.

A meta-query \(Q\) is defined by allowing variables to occur in the class and role positions of conjunctive queries. For example, \(\text{"asking for the relationships that Lucy has to people that are''}\) can be represented as:\n
\[\left\{ \text{p}((\text{Lucy}, ?a)) \land ?a \sqsubseteq \texttt{rdfs\_type} \texttt{person} \right\}

The symbols \(\text{Answer}(Q, K)\) is used to denote the set of all the answers of a meta-query \(Q\) over \(K\).

Reasoning Method Overview

The existing work conclude that under unique name assumption, meta-modeling, i.e., multiple uses of names, can be handled by OWL 2 Punning via renaming, and meta-queries can be answered by translating meta-queries into conjunctive queries via materializing metadata with names.

However, these do not hold anymore in DL-Lite Full due to the presence of \texttt{rdf\_type} in the TBox which makes the KBs to entail extra individual assertions. For example, the axioms and assertions:

\[\texttt{P} \sqsubseteq \texttt{rdfs\_type}, A \sqsubseteq \texttt{rdfs\_type}, \exists B \sqsubseteq \texttt{rdfs\_type}\]

\[\texttt{P}((a, b), (c, e)), \texttt{A}(c, e)\]

cause the extra entailment \(\exists B\texttt{\_on}(\exists C\texttt{\_on}(\exists D\texttt{\_on}(\exists E\texttt{\_on}(a, b))))\) which cannot be computed by OWL 2 Punning, where \(a\) and \(b\) are anonymous element. The extra entailed individual assertions will further affect the results of satisfiability checking and meta-query answering.

For data-layer scalability, the basic idea is to extend the classical query rewriting algorithm \textit{PerfectRef} of DL-Lite\textsubscript{\textit{Q}} to capture the extra knowledge entailed by the non-standard uses of \texttt{rdfs\_type}. For example, for a query atom \(A(x)\) in a query \(Q\), it is not only rewritten as \(B(\texttt{p}(x))\) by the axioms \(\texttt{P} \sqsubseteq \texttt{rdfs\_type}\) to generate new queries, but also rewritten as \(B(\texttt{p}(x))\) by the axioms \(\texttt{P} \sqsubseteq \texttt{rdfs\_type}\) to capture the non-standard uses of \texttt{rdfs\_type}.

The overall extended algorithm \textit{PerfectLit} is shown in Figure 1. For generality, it takes meta-queries as input. In DL-Lite Full, the reduction from satisfiability checking, conjunctive query answering, and meta-query answering to evaluating queries over ABox is realized by this algorithm.

Reasoning Complexity

The above results indicate that satisfiability checking, conjunctive query answering and meta-query answering in DL-Lite Full still have AC\textsubscript{0} data complexity and PTime KB complexity.

Forcoming Research

The future research mainly resides in capturing the non-standard uses of other RDF/S/OWL vocabulary terms as well as optimizing the overall approach of meta-query answering in DL-Lite Full.