Tools for the QuOnto System Conversion between OWL and DL-Lite$_F$ with Protégé - OWL Plug-in

Manuel Dioturni[1], Maurizio Iacovella[2]

[1] Student at University of Rome "La Sapienza", [2] Engineer at Epistematica S.r.l.
manuel_dioturni@hotmail.com, m.iacovella@epistematica.com

1 Introduction

The QuOnto System [1] was developed to implement efficient reasoning algorithms over Knowledge Bases with large amount of instances. It is based on a particular Description Logic, called DL-Lite$_F$ [2], briefly described as:

$$\begin{array}{lcl} Cl & \longrightarrow & A \mid \exists R \mid Cl_1 \sqcap Cl_2 \\ Cr & \longrightarrow & A \mid \exists R \mid \neg A \mid \neg \exists R \\ R & \longrightarrow & P \mid P^- \\ \multicolumn{3}{l}{\text{TBox assertions}: Cl \sqsubseteq Cr, (\mathit{funct}\ R)} \end{array}$$

It represents a good trade-off between expressive power and computational complexity of sound and complete reasoning, in particular, it is tailored to capture basic ontology languages and allows to answer to complex queries expressed on ontologies in LOG-SPACE compared to the data complexity (the data size) [3]. This system doesn't feature yet a user interface nor managing tools and it uses a particular XML syntax to write the ontology, this makes it not easy-to-use. Protégé is an open-source ontology editor based on Java that has a plug-in architecture which makes it flexible and easy to be extended. In addition it can read and write ontologies in OWL format, that is the standard language proposed by W3C. This features have driven the idea to use it as a framework to manage some element of the QuOnto System.

2 Description

The first work for this objective is to translate ontologies written in OWL language to an XML document following the proprietary XML Schema used as TBox input for QuOnto.

The translation from OWL to DL-Lite$_F$ is not always possible because DL-Lite$_F$ is a strict subset of OWL, in particular of OWL-Lite, the least expressive fragment of OWL, which presents some constructs (e.g., some kinds of role restrictions) that cannot be expressed in DL-Lite$_F$, and that make reasoning in OWL-Lite not-tractable if the formalism is not restricted [1]; so it makes the ontology check necessary to know if it is DL-Lite$_F$ expressible, and in this case, we would submit it to the QuOnto engine.

When the ontology has been defined in Protégé [4] it is possible to access its OWL code through the Protégé API [5]. This allows the complete list of the ontology components to be obtained, and subsequently checked and translated.

The ontology check consists of the analysis of the constructs involved in the definition of the particular OWL ontology in order to know whether a corresponding one in DL-Lite$_F$ exists for each one: we use Protégé API to scan the whole ontology structure inspecting each OWL element so to recognize, depending on the nesting of the elements, if the conversion can take place; only in this case we can obtain the corresponding ontology in DL-Lite$_F$ (QuOnto compliant) that now we are able to submit, through the QuOnto API, to the QuOnto reasoner. At this point, it is possible to invoke QuOnto for a consistency check.

This issue has represented the critical task: to design the whole mapping existing between OWL constructs and DL-Lite$_F$ constructs (if the mapping is possible), so to pass by the original ontology in OWL to an equivalent ontology in DL-Lite$_F$, where "equivalent" means that "they have the same logical models".

3 Conclusions and future work

We have shown how it is possible to extend the features of a largely used ontology editor to translate an ontology from OWL to DL-Lite$_F$, in order to carry out reasoning tasks on it using the QuOnto System. The use of the Protégé API has been a valid way to proceed, having it offered useful features to access to the OWL source code. Thanks to this Protégé plug-in, testing the QuOnto capabilities and comparing it with other reasoners will be easier for the user.

In future, we would investigate how to link Protégé directly to the QuOnto Core API to allow to the user to manage this system in a more simple and practical way.

References

1. D.Calvanese, G.De Giacomo, D.Lembo, M.Lenzerini and R.Rosati. Tailoring OWL for Data Intensive Ontologies. In *Proc. of the Workshop on OWL: Experiences and Directions (OWLED 2005), 2005*
2. A.Acciarri, D.Calvanese, G.De Giacomo, D.Lembo, M.Lenzerini, M.Palmieri and R.Rosati. QuOnto: Querying ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, 2005.
3. D.Calvanese, G.De Giacomo, D.Lembo, M.Lenzerini and R.Rosati. DL-Lite: Tractable Description Logics for Ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005), 2005*
4. M.Horridge, H.Knublauch, A.Rector, R.Stevens and C.Wroe. A practical guide to building OWL ontologies using the Protégé-OWL plug-in and CO-ODE tools. *(August 27, 2004)*.
5. H.Knublauch. Protégé OWL-API Programmer's Guide.