From OWL to *DL-Lite* through efficient ontology approximation

Marco Console, Valerio Santarelli, Domenico Fabio Savo

Dipartimento di Ingegneria Informatica Automatica e Gestionale “Antonio Ruberti”
Sapienza Università di Roma
lastname@dis.uniroma1.it

1 Introduction

Ontologies provide a conceptualization of a domain of interest which can be used for different objectives, such as for providing a formal description of the domain of interest for documentation purposes, or for providing a mechanism for reasoning upon the domain. For instance, they are the core element of the Ontology-Based Data Access (OBDA) [3,8] paradigm, in which the ontology is utilized as a conceptual view, allowing user access to the underlying data sources. With the aim to use an ontology as a formal description of the domain of interest, the use of expressive languages proves to be useful. If instead the goal is to use the ontology for reasoning tasks which require low computational complexity, the high expressivity of the language used to model the ontology may be a hindrance. In this scenario, the approximation of ontologies expressed in very expressive languages through ontologies expressed in languages which keep the computational complexity of the reasoning tasks low is pivotal.

In this paper, we focus on the semantic approximation of an ontology for OBDA applications. Thus, we study approaches for approximating ontologies in very expressive languages with ontologies in languages that, characterized by low reasoning complexity, are suitable for query answering purposes. Among the most significant works in which this problem is studied are [7] and [2], in which the described approaches can be traced back to the work of Selman and Kautz [9].

Since OWL 2[1] is the W3C standard language for expressing ontologies, it is often used as the expressive language for formulating ontologies describing the domain of interest. On the other hand, in scenarios in which ontologies are used for OBDA purposes, one naturally focuses on the logics of the *DL-Lite* family [4]. This is a family of Description Logics (DLs) specifically designed to keep all reasoning tasks polynomially tractable in the size of the data, and is thus suitable for OBDA. For this reason, in this work we study the problem of approximating OWL 2 ontologies with ontologies in *DL-Lite*. To this aim we provide an algorithm for the computation of these approximations, and an optimized technique for the computation of the entailment set of an OWL 2 ontology in *DL-Lite*, which can be used efficiently in practice.

[1] http://www.w3.org/TR/2012/REC-owl2-primer-20121211/

2 Approximation of DL ontologies

In this section, we present our notion of approximation of an ontology expressed in a language $\mathcal{L}$ in a target language $\mathcal{L}'$.

We recall that, given a signature Σ and a language $\mathcal{L}$, an $\mathcal{L}$ ontology $\mathcal{O}$ is a set $\mathcal{T} \cup \mathcal{A}$ of assertions over Σ expressed in $\mathcal{L}$, where $\mathcal{T}$, the *TBox*, is a finite set of intentional assertions and $\mathcal{A}$, the *ABox*, is a finite set of instance assertions. Different languages allow for different kinds of TBox and/or ABox assertions and allow for different manners in which these can be combined for obtaining TBoxes and ABoxes in the specific language.

We begin by introducing the notion of *entailment set* [7] of a satisfiable ontology with respect to a language.

Definition 1. *Let $\mathcal{O}$ be a satisfiable ontology expressed in a language $\mathcal{L}$ over a signature Σ, and let $\mathcal{L}$' be a language, not necessarily different from $\mathcal{L}$. The* entailment set *of $\mathcal{O}$ with respect to $\mathcal{L}'$, denoted as $\mathsf{ES}(\mathcal{O}, \mathcal{L}')$, is the set which contains all $\mathcal{L}'$ axioms over Σ that are entailed by $\mathcal{O}$.*

Given an ontology $\mathcal{O}$ and a language $\mathcal{L}'$, we observe that the entailment set of $\mathcal{O}$ with respect to $\mathcal{L}'$ is unique. A straightforward solution in defining the approximation of $\mathcal{O}$ in $\mathcal{L}'$ may be to define this as $\mathsf{ES}(\mathcal{O}, \mathcal{L}')$. This is the solution adopted, for instance, in [7]. Unfortunately, this solution is not suitable for every language, because $\mathsf{ES}(\mathcal{O}, \mathcal{L}')$ may not be a valid $\mathcal{L}'$ ontology. This occurs in two instances. The first is the case in which the entailment set $\mathsf{ES}(\mathcal{O}, \mathcal{L}')$ is infinite. This may happen in *DL-Lite*$_A$, the most expressive DL of the *DL-Lite* family, in which the infiniteness of the entailment set arises from the possibility of inferring infinitely-long existential chains. The second case occurs when $\mathsf{ES}(\mathcal{O}, \mathcal{L}')$ is a finite set of $\mathcal{L}'$-axioms, but, nevertheless, there is no finite set of $\mathcal{L}'$-axioms over the signature of $\mathcal{O}$ that is an $\mathcal{L}'$ ontology and that is logically equivalent to $\mathsf{ES}(\mathcal{O}, \mathcal{L}')$. This may happen when syntactic restrictions are imposed on the manner in which assertions can be combined in order to obtain an ontology in the target language. This is the case for instance for $\mathcal{EL}^{++}$ [1] and *DL-Lite*$_A$.

These observations lead us to formulate the following more sophisticated notion of approximation.

Definition 2. *Let $\mathcal{O}$ be a satisfiable ontology expressed in a language $\mathcal{L}$ over a signature Σ, and let $\mathcal{L}'$ be a language such that $\mathsf{ES}(\mathcal{O}, \mathcal{L}')$ is finite. A satisfiable $\mathcal{L}'$ ontology $\mathcal{O}'$ over Σ is an approximation in $\mathcal{L}'$ of $\mathcal{O}$ if both the following statements hold: (i) $\mathsf{ES}(\mathcal{O}', \mathcal{L}') \subseteq \mathsf{ES}(\mathcal{O}, \mathcal{L}')$; (ii) there is no satisfiable $\mathcal{L}'$ ontology $\mathcal{O}''$ such that $\mathsf{ES}(\mathcal{O}', \mathcal{L}') \subset \mathsf{ES}(\mathcal{O}'', \mathcal{L}') \subseteq \mathsf{ES}(\mathcal{O}, \mathcal{L}')$.*

In other words, a satisfiable ontology $\mathcal{O}'$ is an approximation in $\mathcal{L}'$ of $\mathcal{O}$, if it is an $\mathcal{L}'$ ontology and there is no satisfiable $\mathcal{L}'$ ontology $\mathcal{O}''$ whose entailment set in $\mathcal{L}'$ is "nearer" to the entailment set of $\mathcal{O}$ in $\mathcal{L}'$ than the entailment set in $\mathcal{L}'$ of $\mathcal{O}'$, where the distance here is measured in terms of set inclusion.

It is easy to see that in accordance with Definition 2, there may exist more than one ontology which is an approximation in $\mathcal{L}'$ of $\mathcal{O}$. We denote the set containing these ontologies as $Apx_{MAX}(\mathcal{O}, \mathcal{L}')$.

Algorithm 1: $\mathsf{isApx}(\mathcal{T}, \mathcal{O})$

Input: a $\textit{DL-Lite}_A^{(k)}$ TBox $\mathcal{T}$, a satisfiable OWL 2 ontology $\mathcal{O}$
Output: *true* or *false*
begin
  $\mathcal{E} \leftarrow \mathsf{ES}(\mathcal{T}, \textit{DL-Lite}_A^{(k)})$;
  $\mathcal{S} \leftarrow \mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}) \setminus \mathcal{E}$;
  foreach $\alpha \in \mathcal{S}$
    if $\mathcal{T} \cup \{\alpha\}$ is a $\textit{DL-Lite}_A^{(k)}$ TBox **then return** *false*;
  foreach functionality assertion $\phi \in \mathcal{E}$
    $\mathcal{E} \leftarrow \mathcal{E} \setminus clashes(\phi, \mathcal{E})$;
  foreach functionality assertion $\varphi \in \mathcal{S}$
    if $\mathsf{ES}(\mathcal{E} \setminus clashes(\varphi, \mathcal{E}), \textit{DL-Lite}_A^{(k)}) = \mathsf{ES}(\mathcal{T}, \textit{DL-Lite}_A^{(k)})$ **then return** *false*;
return *true*;
end

3 Approximation in $\textit{DL-Lite}_A$ of OWL 2 ontologies

In this section, we study the problem of computing the approximation of a satisfiable OWL 2 ontology $\mathcal{O}$ with a $\textit{DL-Lite}_A$ TBox. According to Definition 2, to guarantee the existence of an approximation, it is necessary that $\mathsf{ES}(\mathcal{O}, \mathcal{L}')$ be finite. For this reason, in what follows, we only consider versions of $\textit{DL-Lite}_A$, which we denote as $\textit{DL-Lite}_A^{(k)}$, in which only existential chains of bounded length k are allowed in the TBox. As shown in [2], this guarantees that for each $\mathcal{O}$, $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}^{(k)})$ is finite.

We recall that in a $\textit{DL-Lite}_A$ TBox no role (resp. attribute) that is functional or whose inverse is functional can appear positively in the right hand side of a role (resp. attribute) inclusion assertion or in a qualified existential restriction. Now, given a set of $\textit{DL-Lite}_A^{(k)}$ assertions $\mathcal{S}$, and a functionality assertion φ over a role R (resp. attribute U), we denote with $clashes(\varphi, \mathcal{S})$ the set of all assertions involving R (resp. U) that, together with φ, violate the syntactic restriction imposed on $\textit{DL-Lite}_A^{(k)}$ TBoxes. Hence, $clashes(\varphi, \mathcal{S})$ is a set of role (resp. attribute) inclusion assertions and assertions with a qualified existential role (resp. attribute) on the right hand side.

Let $\mathcal{O}$ be an OWL 2 ontology, and let $\mathcal{F}$ be the set containing all the functionality assertions in $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$ for which $clashes(\varphi, \mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})) \neq \emptyset$. If $\mathcal{F} \neq \emptyset$, then $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$ is not a valid $\textit{DL-Lite}_A^{(k)}$ TBox. In what follows, we denote by $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}))$ the set of $\textit{DL-Lite}_A^{(k)}$ TBoxes computed by retracting, from $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$, either $\varphi \in \mathcal{F}$ or the assertions in $clashes(\varphi, \mathcal{S})$, in order to resolve the violations of the syntactic restriction. It is easy to see that, for each $\mathcal{O}$, there are in $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}))$ at most $2^{|\mathcal{F}|}$ TBoxes.

It can be shown that every TBox in $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}))$ satisfies the first condition in Definition 2, and is therefore a candidate for being one of the TBoxes in $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$. However, in order for a TBox

Algorithm 2: computeApx($\mathcal{O}$)

Input: a satisfiable OWL 2 ontology $\mathcal{O}$
Output: a set of $\mathit{DL\text{-}Lite}_A^{(k)}$ TBoxes
begin
  $\mathcal{S} \leftarrow MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)}))$;
  foreach $\mathcal{T}_i \in \mathcal{S}$
    if isApx($\mathcal{T}_i, \mathcal{O}$) $= \mathit{false}$ **then** $\mathcal{S} \leftarrow \mathcal{S} \setminus \{\mathcal{T}_i\}$;
return $\mathcal{S}$;
end

$\mathcal{T}_i$ in $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)}))$ to belong to $Apx_{MAX}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$, it must also satisfy the second condition of Definition 2, and thus that there is no other $\mathit{DL\text{-}Lite}_A^{(k)}$ TBox $\mathcal{T}' \subseteq \mathsf{ES}(\mathcal{T}, \mathit{DL\text{-}Lite}_A^{(k)})$ such that $\mathsf{ES}(\mathcal{T}_i, \mathit{DL\text{-}Lite}_A^{(k)}) \subset \mathsf{ES}(\mathcal{T}', \mathit{DL\text{-}Lite}_A^{(k)}) \subseteq \mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$.

We provide the algorithm isApx which, given a TBox $\mathcal{T}$ and an ontology $\mathcal{O}$, returns *true* if $\mathcal{T} \in Apx_{MAX}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$, *false* otherwise.

With algorithm isApx in place, it is easy to come up with a strategy for computing the approximation in $\mathit{DL\text{-}Lite}^{(k)}$ of an OWL 2 ontology $\mathcal{O}$, that is the one illustrated in algorithm computeApx.

As expected, Algorithm 2 does not return a single TBox, but instead a set of TBoxes. For application purposes, the approximation that shall be used must be chosen from this set. A pragmatic approach could be to choose one non-deterministically. Instead, one could think to leave this choice to the end user, according to the use he intends to make of it. A more interesting direction could be to achieve the identification of a unique TBox by applying some preference criteria to the set returned by Algorithm 2.

The computation of $\mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$ is in general very costly. Indeed, a naive algorithm for computing $\mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$ is the one described in [7], in which: (i) one computes the set Γ of $\mathit{DL\text{-}Lite}_A^{(k)}$ TBox assertions which can be built over the signature of $\mathcal{O}$, and (ii) for each assertion $\alpha \in \Gamma$, such that $\mathcal{O}$ entails α, one adds α to $\mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$. For checking if $\mathcal{O}$ entails α, one needs to use an OWL 2 reasoner.

In the rest of this section we show how to optimize the computation of $\mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$ by providing a technique which drastically reduces in practice the calls to the OWL 2 reasoner.

In the computation of $\mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$, a large portion of the invocations of the OWL 2 reasoner involve assertions in which a general concept $C_{\exists R_1 \ldots \exists R_n}$ involving an existential role chain occurs. Empirical observation, during our tests, has brought to light the fact that this kind of general concept very often does not subsume any concept in $\mathcal{O}$. Hence, all the invocations of the OWL 2 reasoner involving these childless general concepts are useless. Therefore, at the base of our strategy is the identification of all these childless general concepts $C_{\exists R_1 \ldots \exists R_n}$, without invoking the OWL 2 reasoner.

Ontology	# O.A.I.			# N.A.I.			Total time of O.A.I. in ms		
	k = 1	k = 2	k = 3	k = 1	k = 2	k = 3	k = 1	k = 2	k = 3
Pediatric	2.495	2.495	2.495	14.293	78.517	463.861	2.999	2.955	2.992
Mouse Brain	6.059	12.611	19.163	11.018	40.362	157.738	8.426	9.955	12.173
Pathway	10.191	11.999	11.999	14.294	52.374	204.694	11.975	16.498	17.553
Cognitive Atlas	56.883	178.381	474.145	48.006	541.350	6.461.478	348.892	1.812.511	6.832.865
Mammalian Phen.	7.551	7.551	7.551	112.898	413.922	1.618.018	322.527	350.853	350.853
Spatial	51.065	82.735	150.195	47.143	4.541.815	445.019.671	27.827	52.742	132.807

Table 1: Evaluation of the optimization algorithm for the computation of $\mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$. # O.A.I. = number of OWL 2 reasoner invocations by optimized algorithm, # N.A.I. = number of OWL 2 reasoner invocations by non-optimized algorithm.

We will make use of the function $\mathit{subsumed}(S_1, \mathcal{O})$, where S_1 is a general concept (resp. general role, general attribute) which returns the set of atomic concepts (resp. roles, attributes) S_2 such that $\mathcal{O} \models S_2 \sqsubseteq S_1$. This function is efficiently performed by the most commonly-used OWL 2 reasoners, such as Pellet [10], Racer [6], FACT++ [11], and HermiT [5].

Our technique calls, as the first step, for the classification of basic concepts, roles, and attributes, and its encoding into a directed graph, in which the nodes represent the predicates of the ontology, and the edges the inclusion assertions.

After this initial step, the remaining invocations, which we work to minimize, are those needed for computing the entailed inclusion assertions involving general concepts $C_{\exists R_1 \ldots \exists R_n}$, and the entailed disjointness. Regarding the former, we exploit the graph encoding of concept, role, and attribute classification to invoke these subsumption checks in a manner which follows the hierarchical order of these general concepts $C_{\exists R_1 \ldots \exists R_n}$, in order to avoid those checks which can be skipped. Consider, for example, an ontology $\mathcal{O}$ that entails the inclusions $A_1 \sqsubseteq A_2$ and $P_1 \sqsubseteq P_2$, where A_1 and A_2 are concepts and P_1 and P_2 are roles. Exploiting these inclusions we are able to deduce the hierarchical structure involving the general concepts that can be built on these four predicates. For instance, we know that $\exists P_2.A_2 \sqsubseteq \exists P_2$, that $\exists P_2.A_1 \sqsubseteq \exists P_2.A_2$, that $\exists P_1.A_1 \sqsubseteq \exists P_2.A_1$, and so on. We begin by invoking the OWL 2 reasoner by asking for the children of the general concepts which are in the highest position in the hierarchy. So, first we call $\mathit{subsumed}(\exists P_2, \mathcal{O})$. If $\mathit{subsumed}(\exists P_2, \mathcal{O}) = \emptyset$, we then avoid invoking the reasoner asking for $\mathit{subsumed}(\exists P_2.A_2, \mathcal{O})$, and so on. Regarding the latter we follow the same procedure, but beginning from the lowest positions in the hierarchy.

We conclude the computation of $\mathsf{ES}(\mathcal{O}, \mathit{DL\text{-}Lite}_A^{(k)})$ by asking the OWL 2 reasoner for all functionality assertions that are inferred by $\mathcal{O}$.

In Table 1 we present a sample of the evaluation tests for this strategy which we have performed. We have implemented this technique in a Java-based tool and have performed extensive experimentation on a suite of about twenty OWL 2 ontologies that are commonly used as benchmarks for standard ontology reasoning tasks. We present the results of these tests, in which we compare the number of invocations to the OWL 2 reasoner performed with optimizations (O.A.I.), and without (N.A.I.), for computing the entailment set of the OWL 2 ontologies in $DL\text{-}Lite_A^{(k)}$, with $1 \leq k \leq 3$. We also provide, for each ontology, the total time for the computation of $\mathsf{ES}(\mathcal{O}, DL\text{-}Lite_A^{(k)})$.

4 Conclusion

In this paper we have studied the problem of ontology approximation. In particular, we have focused on approximating OWL 2 ontologies with *DL-Lite* TBoxes for OBDA purposes, and presented an optimized technique for this task.

As future work, we plan to improve the performances in computing the approximation in *DL-Lite* of OWL 2 ontologies by adopting more sophisticated techniques. Moreover, it is our intention to study reasonable solutions for addressing the problem of multiple approximations of an ontology, in particular, for those settings in which the approximation is intended to be used for OBDA.

Acknowledgments. This research has been partially supported by the EU under FP7 project Optique – Scalable End-user Access to Big Data (grant n. FP7-318338).

References

1. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope further. In *Proc. of OWLED 2008 DC*, 2008.
2. Elena Botoeva, Diego Calvanese, and Mariano Rodriguez-Muro. Expressive approximations in DL-Lite ontologies. *Proc. of AIMSA 2010*, 2010.
3. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. The Mastro system for ontology-based data access. *Semantic Web J.*, 2(1), 2011.
4. 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. of Automated Reasoning*, 39(3), 2007.
5. Birte Glimm, Ian Horrocks, Boris Motik, Rob Shearer, and Giorgos Stoilos. A novel approach to ontology classification. *J. of Web Semantics*, 10(1), 2011.
6. Volker Haarslev and Ralf Möller. RACER system description. In *Proc. of IJCAR 2001*, volume 2083 of *LNAI*. Springer, 2001.
7. Jeff Z Pan and Edward Thomas. Approximating OWL-DL ontologies. In *Proc. of AAAI 2007*, 2007.
8. Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *J. on Data Semantics*, X, 2008.
9. Bart Selman and Henry Kautz. Knowledge compilation and theory approximation. *J. of the ACM*, 43(2), 1996.
10. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: a practical OWL-DL reasoner. *J. of Web Semantics*, 5(2), 2007.
11. Dmitry Tsarkov and Ian Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of IJCAR 2006*, 2006.