Efficient approximation in *DL-Lite* of OWL 2 ontologies

Marco Console, Valerio Santarelli, Domenico Fabio Savo

Dipartimento di Ingegneria Informatica Automatica e Gestionale "Antonio Ruberti"
SAPIENZA Università di Roma
lastname@dis.uniroma1.it

Abstract. Ontologies, as a conceptualization of a domain of interest, 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 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 of 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 work we present our notion of ontology approximation and present an algorithm for computing the approximation of OWL 2 ontologies by means of DL-Lite TBoxes. Moreover, we provide optimization techniques for this computation, and discuss the results of the implementation of these techniques.

1 Introduction

Ontologies, as a shared conceptualization of a domain of interest, are commonly recognized as powerful tools in support of the information systems of organizations [6,4,10]. On one hand, they can be used for providing a formal description of the domain of interest, and thus represent valuable documentation for an organization. On the other hand, they provide a mechanism for reasoning upon the domain with different objectives. For instance, they are the core element of the Ontology-Based Data Access (OBDA) [13,4] paradigm, in which the ontology is utilized as a conceptual view of the underlying data sources, granting the user the ability to retrieve information without specific knowledge on how this information is organized and where it is stored.

With the aim to use an ontology as a formal description of the domain of interest, the use of expressive languages, such as the OWL 2 Web Ontology Language [9], proves to be useful. The expressivity of such languages allows the ontology designer to obtain a precise formalization of the domain. If instead the goal is to use the ontology for reasoning tasks, the high expressivity of the language used to model the ontology may be of hindrance. In particular, when wishing to access large quantities of data through the ontology, as in OBDA, the computational cost of very expressive languages such as OWL

2 is prohibitive. In these cases, it is necessary to recur to less expressive languages, thus resigning to having less complete representations of the domain of interest.

One possible solution to this issue is to allow the ontology designer the use of expressive languages to define ontologies that model the domain in great detail for the purpose of documentation and of other tasks that do not require strong computational effort, while adopting, for all those reasoning tasks in which particular computational properties are required, such as OBDA, descriptions of the domain of interest obtained through less expressive languages.

Following this approach, the notion of approximation becomes pivotal. The goal of approximation is, given a ontology $\mathcal{O}$ in a language $\mathcal{L}$, to compute an ontology $\mathcal{O}'$ in a target language $\mathcal{L}'$, in which as much as possible of the semantics of $\mathcal{O}$ is preserved. In general, various approaches can be adopted towards this goal [17,14,12,18,3]. Among these, the approaches in which we are most interested are those which aim to obtain the so-called *semantic* approximation [14,3,12]. Here, as opposed to those approaches which aim at a syntactic approximation [17,18], the computation of the ontology which is the approximation of the original one is obtained through the semantics of this original ontology, and not only through its syntactic representation.

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. The most significant works in which this problem is studied are [12] and [3], in which the proposed approaches can be traced back to the work of Selman and Kautz [14].

These approaches, as is ours, are based on the notion of *entailment set*. Given an ontology $\mathcal{O}$ in a language $\mathcal{L}$, the entailment set of $\mathcal{O}$ in a target language $\mathcal{L}'$ is the set of all the assertions expressible in $\mathcal{L}'$ over Σ that are entailed by $\mathcal{O}$. The idea behind our approach is that an approximation in $\mathcal{L}'$ of $\mathcal{O}$ is an ontology $\mathcal{O}'$ whose entailment set minimally differs from the entailment set of $\mathcal{O}$ in $\mathcal{L}'$.

Since OWL 2 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 the context of OBDA, one naturally focuses on the logics of the *DL-Lite* family [5]. This is a family of 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.

The rest of the paper is organized as follows. In Section 2, we provide some useful notions for the paper. In Section 3 we study the problem of ontology approximation, and introduce our notion of approximation. In Section 4 we focus on the approximation in *DL-Lite* of OWL 2 ontologies. In Section 5 we describe our technique for optimizing the computation of the entailment set of an OWL 2 ontology in *DL-Lite*, and present the results of our experimentation. Finally, in Section 6 we conclude the paper.

2 Preliminaries

Description Logics (DLs) [2] are logics that allow one to represent the domain of interests in terms of *concepts*, denoting sets of objects, *value-domains*, denoting sets of values, *attributes*, denoting binary relations between objects and values, and *roles* denoting binary relations over objects.

Let Σ be a signature of symbols for individual (object and value) constants and atomic elements, i.e., concepts, value-domains, attributes, and roles. If $\mathcal{L}$ is a DL, then an ontology $\mathcal{O}$ in $\mathcal{L}$ (or $\mathcal{L}$ ontology) over Σ is the set $\mathcal{T} \cup \mathcal{A}$, where $\mathcal{T}$, the *TBox*, is a finite set of intensional assertions over Σ expressed in $\mathcal{L}$, and $\mathcal{A}$, the *ABox*, is a finite set of instance assertions, i.e, assertions on individuals, over Σ. Different DLs 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 DL.

The semantics of a DL ontology is given in terms of first-order (FOL) interpretations (cf. [2]). We denote with $Mod(\mathcal{O})$ the set of models of $\mathcal{O}$, i.e., the set of FOL interpretations that satisfy all the assertions in $\mathcal{T}$ and $\mathcal{A}$, where the definition of satisfaction depends on the kind of expressions and assertions in the specific DL language in which $\mathcal{O}$ is specified. As usual, a ontology $\mathcal{O}$ is said to be *satisfiable* if it admits at least one model, and $\mathcal{O}$ is said to *entail* a First-Order Logic (FOL) sentence ϕ, denoted $\mathcal{O} \models \phi$, if $\phi^{\mathcal{I}} = \mathit{true}$ for all $\mathcal{I} \in Mod(\mathcal{O})$. Moreover, given to ontology $\mathcal{O}$ and $\mathcal{O}'$, we say that $\mathcal{O}$ and $\mathcal{O}'$ are logically equivalent if $Mod(\mathcal{O}) = Mod(\mathcal{O}')$.

In this work we mainly focus on two specific languages, which are OWL 2, the official ontology language of the World Wide Web Consortium (W3C) [9], and *DL-Lite*$_A$, a member of the *DL-Lite* family [5], which is a family of tractable DLs particularly suited for dealing with ontologies with very large ABoxes, and is at the basis of OWL 2 QL, one of the profiles of OWL 2.

The Web Ontology Language (version 2), or simply OWL 2, is an ontology language for the Semantic Web with formally defined meaning. OWL 2 provides for describing the domain of interest in terms of concepts, roles, attributes, individuals, and data values [9]. Due to the limitation of space, here we do not provide a complete description of OWL 2, but we refer the reader to the official W3C specification [11].

We now present the syntax of the DL *DL-Lite*$_A$. As for the semantics, we apply the general definitions given above, and refer the reader to [13] for the precise description of the semantics of a *DL-Lite*$_A$ ontology.

In what follows we adopt the following notation: A, P, and U are symbols in Σ denoting respectively an atomic concept, an atomic role and an atomic attribute; $T_1, \ldots, T_n$ are n pairwise disjoint unbounded value-domains; B denotes a *basic concept*; C a *general concept*; Q a *basic role*; R a *general role*; V a *general attribute*; E a *basic value-domain*; and F a *value-domain expression*.

Expressions in *DL-Lite*$_A$ are formed according to the following syntax:

$$
\begin{array}{lll}
B \longrightarrow A \mid \exists Q \mid \delta(U) & Q \longrightarrow P \mid P^- & V \longrightarrow U \mid \neg U \\
C \longrightarrow B \mid \neg B \mid \exists Q.C \mid \delta_F(U) & R \longrightarrow Q \mid \neg Q & E \longrightarrow \rho(U) \\
F \longrightarrow T_1 \mid \cdots \mid T_n & &
\end{array}
$$

where: P^- denotes the inverse of P, $\exists Q$, or *unqualified existential restriction* denotes the objects related to some object by the role Q, $\neg$ denotes negation, $\delta(U)$ denotes

the *domain* of U, i.e., the set of objects that U relates to values, and $\rho(U)$ denotes the *range* of U, i.e., the set of values related to objects by U. The concept $\exists Q.C$, or *qualified existential restriction*, denotes the *qualified domain* of Q with respect to C, i.e., the set of objects that Q relates to some instance of C. Similarly, $\delta_F(U)$ denotes the *qualified domain* of U with respect to a value-domain F, i.e., the set of objects that U relates to some value in F. We refer to the qualified existential restriction expression $\exists Q_1.\ldots.\exists Q_n.C$, where C is not a qualified existential restriction, as an existential role chain of depth n.

A *DL-Lite*$_A$ TBox assertion is an assertion of the form:

$$B \sqsubseteq C \qquad Q \sqsubseteq R \qquad E \sqsubseteq F \qquad U \sqsubseteq V \qquad (\mathsf{funct}\ Q) \qquad (\mathsf{funct}\ U)$$

From left to right, the first four assertions denote inclusions between concepts, roles, value-domains, and attributes, respectively. The last two assertions denote functionality on roles and on attributes.

A *DL-Lite*$_A$ TBox is a finite set of assertions of the form above, where suitable limitations in the combination of such assertion are imposed. Given a TBox $\mathcal{T}$ and a role P (resp. an attribute U), we say that P (resp. U) is *primitive* in $\mathcal{T}$ if P does not appear in $\mathcal{T}$ positively in the right-hand side of any role (resp. attribute) inclusion assertion and in any qualified existential restriction. In a *DL-Lite*$_A$ TBox, a role P (resp, attribute U) that is not primitive in $\mathcal{T}$ cannot appear either directly nr inversely in a functionality assertion.

A *DL-Lite*$_A$ ABox $\mathcal{A}$ is a finite set of assertions of the form $A(a)$, $P(a, b)$, and $U(a, v)$, where A, P, and U are as above, a and b are object constants while v is a value constant.

3 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 $\mathcal{L}'$, and then we provide a comparison between our notion and others proposed in literature.

Ontology approximation. In what follows, $\mathcal{O}$ is a satisfiable $\mathcal{L}$ ontology, and $\mathcal{L}'$ a language not necessarily different from $\mathcal{L}$.

First of all, we need to introduce the notion of *entailment set* [12] 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 $\mathcal{O}$.*

In other words, we say that an axiom α belongs to the entailment set of an ontology $\mathcal{O}$ with respect to a language $\mathcal{L}'$, if α is an $\mathcal{L}'$ axiom built over the alphabet of $\mathcal{O}$ and for each interpretation $\mathcal{I} \in Mod(\mathcal{O})$ we have that $\mathcal{I} \models \alpha$. Clearly, given an ontology $\mathcal{O}$ and a language $\mathcal{L}'$, the entailment set of $\mathcal{O}$ with respect to $\mathcal{L}'$ is unique.

With the notion of entailment set in place, we can define the notion of approximation in $\mathcal{L}'$ of $\mathcal{O}'$.

Definition 2. *Let $\mathcal{O}$ be an ontology over a signature Σ. 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, the above definition states that a satisfiable ontology $\mathcal{O}'$ is an approximation in $\mathcal{L}'$ of $\mathcal{O}$, if $\mathcal{O}'$ is an ontology in $\mathcal{L}'$, and there is no a satisfiable ontology $\mathcal{O}''$ in $\mathcal{L}'$ 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.

Given an ontology $\mathcal{O}$, it may be the case that the entailment set in a language $\mathcal{L}'$ of $\mathcal{O}$ is infinite. If so, it may happen that the approximation in $\mathcal{L}'$, in accordance with Definition 2, does not exist. For this reason, we follow the approach proposed in [3], where safeness restrictions on the target language are imposed, in order to guarantee the finiteness of the entailment set. For example, in *DL-Lite*$_A$, which is the language we focus on in the following sections, the infiniteness of the entailment set arises from the possibility of inferring infinitely-long existential chains. In this case, the safeness restriction requires to allow, in the TBox, only existential chains of bounded length. Therefore, in what follows we denote versions of *DL-Lite*$_A$ in which such restriction is enforced as *DL-Lite*$_A^{(k)}$, where k is the bounded length of the existential chains.

Another characteristic of *DL-Lite*$_A$, shared with other languages such as $\mathcal{EL}^{++}$ [1], is that syntactic restrictions are imposed on the manner in which assertions can be combined. In this case, there may exist more than one ontology which is an approximation in $\mathcal{L}'$ of $\mathcal{O}$. In what follows we denote with $Apx_{MAX}(\mathcal{O}, \mathcal{L}')$ the set of $\mathcal{L}'$ ontologies which are approximations in $\mathcal{L}'$ of $\mathcal{O}$ in accordance with Definition 2.

Example 1. Consider the OWL 2 ontology $\mathcal{O} = \{A \sqsubseteq \exists R.B, A \sqsubseteq \forall R.B, (\mathsf{funct}\ R)\}$. It is easy to see that, due to the syntactic restriction in *DL-Lite*$_A$, which imposes that a non primitive role cannot be functional, we have that, up to logical equivalence, the set $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(1)}) = \{\{A \sqsubseteq \exists R.B\}, \{A \sqsubseteq \exists R, (\mathsf{funct}\ R)\}\}$. □

Comparison with related work. The problem of approximating ontologies for OBDA applications has recently been faced in [12] and [3].

In [12], the authors define the approximation in $\mathcal{L}'$ of a satisfiable ontology $\mathcal{O}$ as the set of $\mathcal{L}'$ axioms coinciding with the entailment set of $\mathcal{O}$ with respect to $\mathcal{L}'$.

Definition 3. *Let $\mathcal{O}$ be a satisfiable ontology expressed in a language $\mathcal{L}$. The approximation in $\mathcal{L}'$ of $\mathcal{O}$ is $Apx_{ES}(\mathcal{O}, \mathcal{L}') = \mathsf{ES}(\mathcal{O}, \mathcal{L}')$.*

Therefore, in accordance with the above definition, the approximation in $\mathcal{L}'$ of the ontology $\mathcal{O}$ is a set of $\mathcal{L}'$ axioms, but not necessarily a valid ontology expressed in $\mathcal{L}'$.

In [3], the authors provide a more sophisticated notion of approximation, in which it is required that the approximation in $\mathcal{L}'$ of $\mathcal{O}$ be an ontology expressed in $\mathcal{L}'$.

Definition 4. *Let $\mathcal{O}$ be a satisfiable ontology expressed in a language $\mathcal{L}$ over a signature Σ. 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*) $\mathit{ES}(\mathcal{O}', \mathcal{L}') \subseteq \mathit{ES}(\mathcal{O}, \mathcal{L}')$;
(*ii*) *for each* $\alpha \in \mathit{ES}(\mathcal{O}, \mathcal{L}')$ *such that* $\mathcal{O}' \cup \{\alpha\}$ *is an* $\mathcal{L}'$ *ontology, then* $\alpha \in \mathit{ES}(\mathcal{O}', \mathcal{L}')$.

In other words, an $\mathcal{L}'$ ontology $\mathcal{O}'$ is an approximation in $\mathcal{L}'$ of $\mathcal{O}$ if among all the possible maximal subsets of $\mathsf{ES}(\mathcal{O}, \mathcal{L}')$ which are $\mathcal{L}'$ ontologies, there is one which is logically equivalent to $\mathcal{O}'$. It is not unexpected, as with our approach, that there may exist more than one ontology which is an approximation in $\mathcal{L}'$ of $\mathcal{O}$. We denote with $Apx_{SC}(\mathcal{O}, \mathcal{L}')$ set of $\mathcal{L}'$ ontologies which are all approximations in $\mathcal{L}'$ of $\mathcal{O}$ in accordance with Definition 4.

Differently from Definition 3, Definition 4 guarantees that an approximation is always an ontology expressed in the target language $\mathcal{L}'$.

Similarly to our notion of approximation, if the entailment set is infinite, it may happen that there is no ontology expressed in the target language which is an approximation. Hence, in order to guarantee the existence of an approximation, one must impose safeness restrictions on the target language in this case as well, in order to guarantee the finiteness of the entailment set.

We now compare our approach to those in [3] and [12] by means of some examples.

Example 2. Consider the following OWL 2 ontology: $\mathcal{O} = \{\exists R^- \sqsubseteq B,\ A \sqsubseteq \exists R.B, (\mathsf{funct}\ R)\}$. In accordance with Definitions 3, 4, 2, we have:

$$Apx_{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(1)}) = \{\exists R^- \sqsubseteq B,\ A \sqsubseteq \exists R,\ A \sqsubseteq \exists R.B,\ A \sqsubseteq \exists R.\exists R^-,\ \exists R \sqsubseteq \exists R.\exists R^-,\ \exists R^- \sqsubseteq \exists R^-.\exists R,\ (\mathsf{funct}\ R)\ \}$$

$$Apx_{SC}(\mathcal{O}, \textit{DL-Lite}_A^{(1)}) = \{\mathcal{O}'_{sc} = \{\exists R^- \sqsubseteq B,\ A \sqsubseteq \exists R,\ A \sqsubseteq \exists R.B, A \sqsubseteq \exists R.\exists R^-,\ \exists R \sqsubseteq \exists R.\exists R^-,\ \exists R^- \sqsubseteq \exists R^-.\exists R\ \},\ \mathcal{O}''_{sc} = \{\exists R^- \sqsubseteq B,\ A \sqsubseteq \exists R,\ \ (\mathsf{funct}\ R)\ \}\ \}$$

$$Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(1)}) = \{\mathcal{O}_{max} = \{\exists R^- \sqsubseteq B,\ A \sqsubseteq \exists R,\ \ (\mathsf{funct}\ R)\ \}\ \} \qquad \square$$

Due to the syntactic restrictions enforced in *DL-Lite*$_A^{(1)}$, ontology $\mathcal{O}$ of Example 2 is not a *DL-Lite*$_A^{(1)}$ ontology. However, it is logically equivalent to the *DL-Lite*$_A^{(1)}$ ontology $\{\exists R^- \sqsubseteq B, A \sqsubseteq \exists R, (\mathsf{funct}\ R)\}$.

Regarding $Apx_{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(1)})$ we only observe that it is not a valid *DL-Lite*$_A^{(1)}$ ontology. Up to logical equivalence, we can see that the set $Apx_{SC}(\mathcal{O}, \textit{DL-Lite}_A^{(1)})$ contains, along with ontology $\mathcal{O}''_{sc}$, which is logically equivalent to $\mathcal{O}$, also the unexpected ontology $\mathcal{O}'_{sc}$, for which we have $\mathsf{ES}(\mathcal{O}'_{sc}, \textit{DL-Lite}_A^{(1)}) \subset \mathsf{ES}(\mathcal{O}''_{sc}, \textit{DL-Lite}_A^{(1)})$. Finally, according to Definition 2, up to logical equivalence, the only approximation is a *DL-Lite*$_A^{(1)}$ ontology that is logically equivalent to $\mathcal{O}$.

Other significant differences between these three approaches arise when the goal is to compute the approximation of an ontology $\mathcal{O}$ expressed in a language $\mathcal{L}$ in which the UNA is not adopted into a target language $\mathcal{L}'$ in which the UNA is adopted. In the example below, we highlight the different behavior of the three approaches in this circumstance.

Example 3. We recall that in OWL 2 the UNA is not adopted. This means, for instance, that given the following two ontologies $\mathcal{O} = \{A \sqsubseteq \{o_1\}, B \sqsubseteq \{o_2\}\}$ and $\mathcal{O}' = \{A \sqsubseteq$

$\{o_1\}, B \sqsubseteq \{o_2\}, o_1 \neq o_2\}$, we have that $Mod(\mathcal{O}) \neq Mod(\mathcal{O}')$. In fact, while $\mathcal{O}$ does not entail $A \sqsubseteq \neg B$, $\mathcal{O}'$ does. Differently, if we adopt the UNA in OWL 2, then we have that $Mod(\mathcal{O}) = Mod(\mathcal{O}')$, and both entail the assertion $A \sqsubseteq \neg B$.

From the observations above, it follows that $\{A \sqsubseteq \{o_1\}, B \sqsubseteq \{o_2\}\} \subseteq \mathsf{ES}(\mathcal{O}, \text{OWL}_{UNA})$, and that $A \sqsubseteq \neg B \notin \mathsf{ES}(\mathcal{O}, \text{OWL}_{UNA})$. In what follows let us denote with OWL_{UNA} the version of OWL 2 where the UNA is adopted.

In accordance with Definitions 3, 4, and 2, we have that, up to logical equivalence, the approximations in OWL_{UNA} of $\mathcal{O}$ are:

- $Apx_{ES}(\mathcal{O}, \text{OWL}_{UNA}) = \mathsf{ES}(\mathcal{O}, \text{OWL}_{UNA})$. In this case, $Apx_{ES}(\mathcal{O}, \text{OWL}_{UNA})$ is a valid OWL_{UNA} ontology. Let $\mathcal{O}_{ES}$ be such ontology. As shown above, since in OWL_{UNA} the UNA is adopted, $\mathcal{O}_{ES} \models A \sqsubseteq \neg B$. This means that $\mathsf{ES}(\mathcal{O}_{ES}, \text{OWL}_{UNA}) \not\subseteq \mathsf{ES}(\mathcal{O}, \text{OWL}_{UNA})$.
- $Apx_{SC}(\mathcal{O}, \text{OWL}_{UNA}) = \emptyset$. Indeed, since $\mathsf{ES}(\mathcal{O}, \text{OWL}_{UNA})$ is a valid OWL_{UNA} ontology, any OWL_{UNA} ontology $\mathcal{O}_1$ such that $\mathsf{ES}(\mathcal{O}_1, \text{OWL}_{UNA}) \subset \mathsf{ES}(\mathcal{O}, \text{OWL}_{UNA})$, does not satisfy condition (ii) of Definition 4. Moreover, since every OWL_{UNA} ontology $\mathcal{O}_2$ that satisfies condition (ii) entails both $\{A \sqsubseteq \{o_1\}, B \sqsubseteq \{o_2\}\}$, then it also entails $A \sqsubseteq \neg B$. Hence $\mathsf{ES}(\mathcal{O}_2, \text{OWL}_{UNA}) \not\subseteq \mathsf{ES}(\mathcal{O}, \text{OWL}_{UNA})$, and so condition (i) of Definition 4 is not satisfied.
- $Apx_{MAX}(\mathcal{O}, \text{OWL}_{UNA}) = \{\mathcal{O}'_{max} = \{A \sqsubseteq \{o_1\}\}, \mathcal{O}''_{max} = \{B \sqsubseteq \{o_2\}\}\}$. Indeed, it is easy to verify that both $\mathcal{O}'_{max}$ and $\mathcal{O}''_{max}$ satisfy both conditions (i) and (ii) of Definition 2, and that there is no other ontology $\mathcal{O}'''$ in $Apx_{MAX}(\mathcal{O}, \text{OWL}_{UNA})$ logically equivalent to neither $\mathcal{O}'_{max}$ nor $\mathcal{O}''_{max}$. □

As shown in the previous example, given an $\mathcal{L}$ ontology $\mathcal{O}$ and a target language $\mathcal{L}'$, our approach, as the one in [3], but differently from the one given in [12], guarantees that an approximation in $\mathcal{L}'$ of $\mathcal{O}$ is an $\mathcal{L}'$ ontology, and that for each $\mathcal{L}'$ ontology $\mathcal{O}'$ that is an approximation in $\mathcal{L}'$ of $\mathcal{O}$, we have that $\mathsf{ES}(\mathcal{O}', \mathcal{L}') \subseteq \mathsf{ES}(\mathcal{O}, \mathcal{L}')$. Finally, it can be shown that our approach always preserves the same or more inferences than those obtained by adopting the approach given in [3].

Theorem 1. *For each $\mathcal{O}_{sc} \in Apx_{SC}(\mathcal{O}, \mathcal{L}')$, there is an $\mathcal{O}_{max} \in Apx_{MAX}(\mathcal{O}, \mathcal{L}')$ such that $\mathsf{ES}(\mathcal{O}_{sc}, \mathcal{L}') \subseteq \mathsf{ES}(\mathcal{O}_{max}, \mathcal{L}')$. Furthermore, for each $\mathcal{O}_{max} \in Apx_{MAX}(\mathcal{O}, \mathcal{L}')$, there is no $\mathcal{O}_{sc} \in Apx_{SC}(\mathcal{O}, \mathcal{L}')$ such that $\mathsf{ES}(\mathcal{O}_{max}, \mathcal{L}') \subset \mathsf{ES}(\mathcal{O}_{sc}, \mathcal{L}')$.*

4 Approximation in *DL-Lite*$_A$ of OWL 2 ontologies

In this section, we study the problem of computing the approximation in *DL-Lite*$_A$ of a satisfiable OWL 2 ontology $\mathcal{O}$. More specifically, we aim to approximate $\mathcal{O}$ with *DL-Lite*$_A$ TBox assertions. Therefore, in what follows we assume that $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A)$ is a set of *DL-Lite*$_A$ TBoxes and that $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A)$ is a set of *DL-Lite*$_A$ TBox assertions. To guarantee the finiteness of the entailment set, we refer to versions of *DL-Lite*$_A$ allowing only for TBox assertions with existential chains of bounded length k.

Given a set of *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 *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 *DL-Lite*$_A^{(k)}$ TBox, and there are at most $2^{|\mathcal{F}|}$ TBoxes $\mathcal{T}_i$ which are valid *DL-Lite*$_A^{(k)}$ TBoxes and minimally differ from $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$. Let $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}))$ be the set of such *DL-Lite*$_A^{(k)}$ TBoxes. One can compute these TBoxes in $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}))$ 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.

The lemma below guarantees that a TBox in $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}))$ is a candidate for being one of the TBoxes in $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$.

Lemma 1. *Let $\mathcal{O}$ be a satisfiable OWL 2 ontology and let $\mathcal{T}$ be a DL-Lite*$_A^{(k)}$ *TBox. $\mathsf{ES}(\mathcal{T}, \textit{DL-Lite}_A^{(k)}) \subseteq \mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$ if and only if $\mathcal{T} \subseteq \mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$.*

In other words, the above lemma guarantees that the first condition in Definition 2 is satisfied by every *DL-Lite*$_A^{(k)}$ TBox in $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}))$, and that in computing all the TBoxes in $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$, one can consider only the assertions in $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$. Moreover, from the monotonicity of the DLs, it directly follows that for every TBox $\mathcal{T}$ in $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$ there is a TBox in $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}))$ which is logically equivalent to $\mathcal{T}$.

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

To explain why a TBox in $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)}))$ does not necessarily also satisfy the second condition in Definition 2, we refer to the ontology $\mathcal{O} = \{\exists R^- \sqsubseteq B, A \sqsubseteq \exists R.B, (\mathsf{funct}\ R)\}$ of Example 2. It is easy to see that $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(1)}) = \{\exists R^- \sqsubseteq B, A \sqsubseteq \exists R.B, (\mathsf{funct}\ R), \exists R \sqsubseteq \exists R.\exists R^-, \exists R \sqsubseteq \exists R.B, A \sqsubseteq \exists R.\exists R^-, \exists R^- \sqsubseteq \exists R^-.\exists R, A \sqsubseteq \exists R\}$, and that $clashes((\mathsf{funct}\ R), \mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(1)})) = \{A \sqsubseteq \exists R.B, \exists R \sqsubseteq \exists R.\exists R^-, \exists R \sqsubseteq \exists R.B, A \sqsubseteq \exists R.\exists R^-, \exists R^- \sqsubseteq \exists R^-.\exists R\}$. Hence, by following the procedure described above, we have the following two TBoxes in $MaxSub_{ES}(\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(1)}))$: $\mathcal{T}_1 = \{\exists R^- \sqsubseteq B, A \sqsubseteq \exists R.B, \exists R \sqsubseteq \exists R.\exists R^-, \exists R \sqsubseteq \exists R.B, A \sqsubseteq \exists R.\exists R^-, \exists R^- \sqsubseteq \exists R^-.\exists R, A \sqsubseteq \exists R\}$, and $\mathcal{T}_2 = \{\exists R^- \sqsubseteq B, (\mathsf{funct}\ R), A \sqsubseteq \exists R\}$. However, it is clear that only $\mathcal{T}_2 \in Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(1)})$. In fact we have that $\mathsf{ES}(\mathcal{T}_1, \textit{DL-Lite}_A^{(1)}) \subset \mathsf{ES}(\mathcal{T}_2, \textit{DL-Lite}_A^{(1)})$.

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

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

Input: a *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 *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

lows. Given a satisfiable OWL 2 ontology $\mathcal{O}$ and a *DL-Lite*$_A^{(k)}$ TBox $\mathcal{T}$, it first computes their entailment sets in *DL-Lite*$_A^{(k)}$ and then computes the set $\mathcal{S}$ containing all the assertions in the entailment set of $\mathcal{O}$ in *DL-Lite*$_A^{(k)}$ which are not in the entailment set of $\mathcal{T}$ in *DL-Lite*$_A^{(k)}$. Then, for every assertion α in $\mathcal{S}$, it attempts to add α to $\mathcal{T}$ without violating any syntactic restriction. If such assertion exists, then $\mathcal{T}$ is not an approximation in *DL-Lite*$_A^{(k)}$ of $\mathcal{O}$. If not, the algorithm continues, fixing $\mathcal{E}$ as the entailment set of $\mathcal{T}$ in *DL-Lite*$_A^{(k)}$, and resolving every violation of the syntactic restriction in $\mathcal{E}$ by removing $clashes(\phi, \mathcal{E})$ from $\mathcal{E}$, for every functionality assertion ϕ in $\mathcal{E}$. This operation guarantees that $\mathcal{E}$ is a *DL-Lite*$_A^{(k)}$ TBox. Then the algorithm checks, for every functionality assertion φ in $\mathcal{S}$, if the set obtained from $\mathcal{E}$ by removing $clashes(\varphi, \mathcal{E})$ from $\mathcal{E}$ is logically equivalent to $\mathcal{E}$. If this is the case, then it is possible to build a *DL-Lite*$_A^{(k)}$ TBox $\mathcal{T}''$ by adding φ to the set of assertions obtained in this fashion. $\mathcal{T}''$ is a *DL-Lite*$_A^{(k)}$ TBox that proves that $\mathcal{T}$ does not satisfy the second condition of Definition 2. Otherwise, the algorithm terminates by returning *true*.

The theorem below establishes termination and correctness of Algorithm 1.

Theorem 2. *Let $\mathcal{O}$ be a satisfiable OWL 2 ontology, and let $\mathcal{T}$ be a DL-Lite*$_A^{(k)}$ *TBox. $isApx(\mathcal{T}, \mathcal{O})$ terminates, and returns true if and only if $\mathcal{T} \in Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$.*

Given a satisfiable OWL 2 ontology $\mathcal{O}$, Lemma 1 and Theorem 2 suggest Algorithm 2 for computing, up to logical equivalence, the set $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$.

The following theorem establishes termination and correctness of Algorithm 2.

Theorem 3. *Let $\mathcal{O}$ be a satisfiable OWL 2 ontology. Then $computeApx(\mathcal{O})$ terminates and computes, up to logical equivalence, $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$.*

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

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

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

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.

5 Efficient entailment set computation in *DL-Lite*

In the previous sections, we have provided an algorithm for computing the set $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$ of TBoxes for an OWL 2 ontology $\mathcal{O}$. This computation is clearly intractable. Indeed, it requires to compute the set $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$, and moreover, due to the syntactic restriction enforced in *DL-Lite*$_A^{(k)}$, in the worst case, the cardinality of the set $Apx_{MAX}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$ is exponential in the number of functionality assertions in $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$. In particular, the computation of $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$ is in general very costly, as highlighted also in [3] and [12], since it requires the invocation of reasoning services over an OWL 2 ontology $\mathcal{O}$. This task is performed by invoking an OWL 2 oracle which can be implemented by an OWL 2 reasoner.

A naive algorithm for computing $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$ is the one described in [12], in which firstly one computes the set Γ of *DL-Lite*$_A^{(k)}$ TBox assertions which can be built over the signature Σ, and then, for each assertion $\alpha \in \Gamma$ such that $\mathcal{O} \models \alpha$, one adds α to $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$.

We now show how to optimize the computation of $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$ through a technique which drastically reduces in practice the calls to the OWL 2 oracle.

In the computation of $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$, a large portion of the invocations of the OWL 2 oracle involve assertions in which a general concept $C_{\exists R_1 \ldots \exists R_n}$ involving a non trivial existential role chain occurs. Empirical observation 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 oracle 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 oracle.

We use the function $subsumed(S_1, O)$, where S_1 is a general concept (resp. general role, general attribute) which returns the set of concepts (resp. roles, attributes) S_2 such that $\mathcal{O} \models \mathcal{S}_2 \sqsubseteq S_1$. This function is efficiently performed by the most commonly-used OWL 2 reasoners, such as Pellet [15], Racer [8], FACT++ [16], and HermiT [7].

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

Ontology	DL Fragment	#O.A.			#N.A.			Total computation time (ms)		
		k = 1	k = 2	k = 3	k = 1	k = 2	k = 3	k = 1	k = 2	k = 3
Mouse	$\mathcal{ALCI}$	6.059	12.611	19.163	11.018	40.362	157.738	8.426	9.955	12.173
Pathway	$\mathcal{EL}$	10.191	11.999	11.999	14.294	52.374	204.694	11.975	16.498	17.553
Cognitive	$\mathcal{SHIF}(D)$	56.883	178.381	474.145	48.006	541.350	6.461.478	348.892	1.812.511	6.832.865
Mammalian	$\mathcal{AL}+$	7.551	7.551	7.551	112.898	413.922	1.618.018	322.527	350.853	350.853
Spatial	$\mathcal{ALEHI}$	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 optimized algorithm for computing $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$.

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 can deduce the hierarchical structure involving 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 oracle, asking for the children of the general concepts which are in the highest position in the hierarchy. So, first we call $subsumed(\exists P_2, \mathcal{O})$. If $subsumed(\exists P_2, \mathcal{O}) = \emptyset$, we avoid invoking the oracle asking for $subsumed(\exists P_2.A_2, \mathcal{O})$, and so on. Regarding the latter we follow the same procedure, but beginning from the lowest positions in the hierarchy. The algorithm concludes by asking the OWL 2 oracle for all functionalities entailed by $\mathcal{O}$.

In Table 1 we present a sample of the evaluation tests for this strategy, performed through a Java-based implementation of this technique. The table reports the number of invocations to the OWL 2 oracle performed with our optimization (#O.A.), and without (#N.A.), for computing the entailment set of the ontologies in $\textit{DL-Lite}_A^{(k)}$, with $1 \leq k \leq 3$. It also reports, for each ontology, the total computation time of $\mathsf{ES}(\mathcal{O}, \textit{DL-Lite}_A^{(k)})$.

6 Conclusion

In this paper we address the problem of ontology approximation. We illustrate our approach to this problem, and provide a comparison with other approaches provided in literature. We deeply investigate the approximation of OWL 2 ontologies with *DL-Lite* TBoxes for OBDA purposes, and provide an algorithm for its computation. Finally, we present a technique for the optimization of the core procedure of this computation, whose success we have shown with empirical evaluation. 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, we aim to study reasonable solutions for addressing the problem of multiple approximations of an ontology. In particular, for those settings in which the approximation is used in OBDA.

Acknowledgments. This research has been partially supported by the EU under FP7 project Optique (grant n. FP7-318338), and by the EU under FP7-ICT project ACSI (grant no. 257593).

References

1. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope further. In *Proc. of OWLED 2008 DC*, 2008.
2. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
3. Elena Botoeva, Diego Calvanese, and Mariano Rodriguez-Muro. Expressive approximations in DL-Lite ontologies. *Proc. of AIMSA 2010*, pages 21–31, 2010.
4. 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):43–53, 2011.
5. 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):385–429, 2007.
6. Balakrishnan Chandrasekaran, John R Josephson, and V Richard Benjamins. What are ontologies, and why do we need them? *Intelligent Systems and Their Applications, IEEE*, 14(1):20–26, 1999.
7. Birte Glimm, Ian Horrocks, Boris Motik, Rob Shearer, and Giorgos Stoilos. A novel approach to ontology classification. *J. of Web Semantics*, 10(1), 2011.
8. Volker Haarslev and Ralf Möller. RACER system description. In *Proc. of IJCAR 2001*, volume 2083 of *LNAI*, pages 701–705. Springer, 2001.
9. Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph, editors. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation, 11 december 2012. Available at http://www.w3.org/TR/2012/REC-owl2-primer-20121211/.
10. Maurizio Lenzerini. Ontoloogy-based data management. In *Proc. of CIKM 2011*, pages 5–6, 2011.
11. Boris Motik, Bijan Parsia, and Peter F. Patel-Schneider, editors. *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax*. W3C Recommendation, 11 december 2012. Available at http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/.
12. Jeff Z Pan and Edward Thomas. Approximating OWL-DL ontologies. In *Proc. of AAAI 2007*, page 1434, 2007.
13. Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.
14. Bart Selman and Henry Kautz. Knowledge compilation and theory approximation. *J. of the ACM*, 43(2):193–224, 1996.
15. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: a practical OWL-DL reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53, 2007.
16. Dmitry Tsarkov and Ian Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of IJCAR 2006*, pages 292–297, 2006.
17. Tuvshintur Tserendorj, Sebastian Rudolph, Markus Krötzsch, and Pascal Hitzler. Approximate OWL-reasoning with Screech. In *Proc. of RR 2008*, pages 165–180. Springer, 2008.
18. Holger Wache, Perry Groot, and Heiner Stuckenschmidt. Scalable instance retrieval for the semantic web by approximation. In *Proc. of WISE 2005*, pages 245–254. Springer, 2005.