Expressive Approximations in *DL-Lite* Ontologies ⋆

Elena Botoeva, Diego Calvanese, and Mariano Rodriguez-Muro

KRDB Research Centre
Free University of Bozen-Bolzano, Italy
{botoeva,calvanese,rodriguez}@inf.unibz.it

Abstract. Ontology based data access (OBDA) is concerned with providing access to typically very large data sources through a mediating conceptual layer that allows one to improve answers to user queries by taking into account domain knowledge. In the context of OBDA applications, an important issue is that of reusing existing domain ontologies. However, such ontologies are often formulated in expressive languages, which are incompatible with the requirements of efficiently accessing large amounts of data. Approximation of such ontologies by means of less expressive ones has been proposed as a possible solution to this problem. In this work we present our approach to semantic (as opposed to syntactic) approximation of OWL 2 TBoxes by means of TBoxes in *DL-Lite*$_{\mathcal{A}}$. The point of interest in *DL-Lite*$_{\mathcal{A}}$ approximations is capturing entailments involving chains of existential role restrictions, which can play an essential role in query answering. The presence of TBox assertions involving existential chains affects query answering by enriching the number of obtained rewritings, and hence allows us to cope better with incomplete information about object and data properties. We provide an approximation algorithm and show its soundness and completeness. We also discuss the implementation of the algorithm.

1 Introduction

Ontology based data access (OBDA) [7,15,4] is concerned with providing access to typically large data sources through a conceptual layer constituted by an ontology. Such a mediating ontology on the one hand provides a high level conceptual view of the data residing at the sources, thus freeing users from the need to be aware of the precise organization of the data, and on the other hand allows for improving answers to user queries by taking into account the domain knowledge encoded in the ontology.

In the context of OBDA applications, an important issue is that of reusing existing domain ontologies. However, such ontologies are designed to be as general as possible, and hence are often formulated in expressive languages, such as the Web Ontology Language OWL 2 [6], that are incompatible with the requirements of efficiently accessing large amounts of data. Given the well known trade-off of language expressiveness vs. complexity of reasoning in ontology languages [3], in order to regain efficiency of inference in data access, it might be necessary to *approximate* an ontology formulated in an expressive language by means of an ontology formulated in a less expressive

⋆ This work has been partially supported by the ICT Collaborative Project ACSI, funded by the EU under FP7 grant agreement n. 257593.

language that exhibits nice computational properties, in particular with respect to data complexity. Examples of such languages are those of the $\mathcal{EL}$ family [2,11], and those of the *DL-Lite* family [5,4], where the latter languages have been designed specifically for allowing efficient access to large amounts of data. The problem of computing approximations of ontologies in OWL 2 (or its expressive fragments) has been addressed considering languages of both families as target [13,14,17].

An approximation should be as faithful as possible, i.e., capture as much as possible of the semantics of the original ontology. A basic requirement is that the approximation is *sound*, i.e., that it does not imply additional unwanted inferences. Instead, *completeness* guarantees that all entailments of the original ontology that are also expressible in the target language are preserved in the approximated ontology. A common type of approximations are syntactic approximations, which are transformation of the original ontology that only consider the syntactic form of the axioms that are to be approximated [19]. This kind of approximation generally allows for fast and simple algorithms, however in general it does not guarantee soundness and/or completeness. More interesting are *semantic approximations*, which exploit the semantics of the original ontology to compute the approximated one [18]. Algorithms for computing this kind of approximations tend to be slower since they often involve sound and complete reasoning in the expressive language of the original ontology, e.g., to perform a complete classification of the concepts. However, they can also provide better guarantees with respect to soundness and completeness of the result. Selman and Kautz [18] introduced the term knowledge compilation for computing such approximations.

In this paper we focus on sound and complete semantic approximations of OWL 2 ontologies by means of *DL-Lite* ontologies: specifically, as target ontology language we consider *DL-Lite*$_{\mathcal{A}}$, which is an expressive member of the *DL-Lite* family [15,4] known to have very nice computational properties. Moreover, polynomial reasoning techniques for this logic were developed and implemented in the system QuOnto [1,16].

For the purpose of approximation of OWL 2 in *DL-Lite*$_{\mathcal{A}}$, it suffices to consider only the TBox of an ontology, which represents the intensional information about the domain of interest. Indeed, the extensional knowledge is represented by data sources that are accessed through the TBox of the ontology. The objective is to compile, in the best possible way, the knowledge expressed in the OWL 2 TBox into a *DL-Lite*$_{\mathcal{A}}$ TBox approximation. The latter can then be used by application designers in scenarios where they need to access large amounts of data (i.e., ABoxes), and in which reasoning over the original, expressive ontology would be practically unfeasible.

Our work represents an important extension of previous work on semantic approximation in the *DL-Lite* family [13], which proposes an algorithm that approximates OWL DL ontologies in *DL-Lite*$_{\mathcal{F}}$. A crucial difference between *DL-Lite*$_{\mathcal{F}}$ and *DL-Lite*$_{\mathcal{A}}$ is that the former, but not the latter, rules out role hierarchies, and nested qualified existentials on the right-hand part of concept inclusion assertions [15,4]. On the one hand, this added expressive power is of importance in applications [10]. On the other hand it makes the task of computing sound and complete semantic approximations significantly more challenging. Indeed, while for *DL-Lite*$_{\mathcal{F}}$ the number of different concepts that can be expressed with a given finite alphabet of concept (i.e., unary relation) and role (i.e., binary relation) symbols is finite, this is not the case for *DL-Lite*$_{\mathcal{A}}$, due to the

presence of concepts of the form $\exists R_1 . \ldots \exists R_n . A$, where $R_1, \ldots, R_n$ are roles and A is a concept name. We call such concepts making use of nested qualified existentials *existential role chains*. They can be used in the rewriting step of query answering algorithms (see, e.g., [5]), and hence they play an essential role in query answering, as illustrated by the following example.

Example 1. Consider the medical OWL 2 TBox containing the following:

$$\begin{array}{rcl@{\qquad}rcl}
\mathit{Pharyngitis} & \sqsubseteq & \exists \mathit{hasTreatment}.\{\mathsf{ciproxin}\} & \{\mathsf{ciproxin}\} & \sqsubseteq & \mathit{Antibiotic} \\
\{\mathsf{ciproxin}\} & \sqsubseteq & \forall \mathit{sideEffect}.(\mathit{Nausea} \sqcup \mathit{RenalPain}) & \mathit{Antibiotic} & \sqsubseteq & \mathit{Drug} \\
\mathit{Nausea} & \sqsubseteq & \exists \mathit{symptomOf}.\mathit{AcuteRenalFailure} & \mathit{Drug} & \sqsubseteq & \exists \mathit{sideEffect} \\
\mathit{RenalPain} & \sqsubseteq & \exists \mathit{symptomOf}.\mathit{AcuteRenalFailure} & & &
\end{array}$$

Assume that an ABox, built from a large database containing patient records, contains the assertions $\mathit{Pharyngitis}(\mathsf{c})$ and $\mathit{hasCondition}(\mathsf{john}, \mathsf{c})$. Consider a clinical trial query asking for patients that have a treatment that might cause or be involved in acute renal failure symptoms:

$$\begin{aligned} q(x) \leftarrow\ & \mathit{hasCondition}(x, y), \mathit{hasTreatment}(y, z), \mathit{sideEffect}(z, m), \\ & \mathit{symptomOf}(m, n), \mathit{AcuteRenalFailure}(n) \end{aligned}$$

Trying to answer the query with an OWL 2 reasoner might fail due to the amount of data in the ABox and the complexity of the query. On the other hand, approximating syntactically the ontology or by means of the algorithm in [13] will fail to give the expected answer, i.e., $\{\mathsf{john}\}$ because the entailment $\mathit{Pharyngitis} \sqsubseteq \exists \mathit{hasTreatment}.\exists \mathit{sideEffect}.\exists \mathit{symptomOf}.\mathit{AcuteRenalFailure}$ is not captured. To capture such an entailment, a form of approximation taking into account existential role chains is required. ∎

Approaching the problem is non-trivial if one wants to keep soundness and completeness of the approximation. On the one hand, entailments involving existential chains can come from complex OWL 2 concept descriptions, as we have seen in the example. On the other hand, there is no a priori bound on the length of the chains that have to be considered. In this paper we show that by suitably extending the alphabet, it is possible to capture all *DL-Lite*$_\mathcal{A}$ entailments involving existential chains (of arbitrary length). We also show that it is not possible to capture all *DL-Lite*$_\mathcal{A}$ entailments if the alphabet is not extended; however, we propose a compromise on the length of the entailed formulas that provides useful guarantees of completeness. We demonstrate the proposed approach in a Java based, open source, approximation engine that is available as a Java library, a command line OWL 2-to-*DL-Lite*$_\mathcal{A}$ approximation tool, and a Protege 4.0 plugin.

2 Preliminaries on Description Logics

Description Logics (DLs) [3] are logics specifically designed for representing structured knowledge, and they provide the formal underpinning for the standard ontology languages OWL and OWL 2 [6]. We introduce now OWL 2 and *DL-Lite*$_\mathcal{A}$, the two DLs that we deal with in this paper.

OWL 2. The Web Ontology language[1] OWL 2 is an ontology language for the Semantic Web that has been designed by W3C to represent rich and complex knowledge and to reason about it. OWL 2 corresponds to the DL $\mathcal{SROIQ}$ [8], that we now define.[2]

In DLs, the domain of interest is modeled by means of concepts and roles, denoting respectively unary and binary predicates. The language of OWL 2 contains atomic concept names A, atomic role names P, and individual names a. *Complex concepts* and *roles*, denoted respectively by C and R, are defined as:

$$R ::= P_\top \mid P_\bot \mid P \mid P^- \qquad C ::= \top \mid \bot \mid A \mid \neg C \mid C_1 \sqcap \cdots \sqcap C_n \mid C_1 \sqcup \cdots \sqcup C_n \mid \forall R.C \mid \exists R.C \mid \exists R.\mathsf{Self} \mid \geq k\, R.C \mid \leq k\, R.C \mid \{a_1, \ldots, a_n\}$$

A concept of the form $\exists R.C$ is called a *qualified existential (restriction)*, and the simpler form $\exists R.\top$, in the following abbreviated as $\exists R$, is called an *unqualified existential*.

In DLs, the intensional knowledge about the domain is represented in a TBox, consisting of a finite set of axioms and constraints involving concepts and roles. An OWL 2 *TBox*, $\mathcal{T}$, is a finite set of:

(i) *concept inclusion axioms* of the form $C_1 \sqsubseteq C_2$,
(ii) *role inclusion axioms* of the form $R_1 \circ \cdots \circ R_n \sqsubseteq R$, $n \geq 1$, and
(iii) *role constraints*, such as disjointness, functionality, transitivity, asymmetry, symmetry, irreflexivity, and reflexivity, expressed respectively with $\mathsf{Dis}(R_1, R_2)$, $\mathsf{Fun}(R)$, $\mathsf{Trans}(P)$, $\mathsf{Asym}(P)$, $\mathsf{Sym}(P)$, $\mathsf{Irr}(P)$, and $\mathsf{Ref}(P)$.

Note that some of the role constraints can be expressed using concept or role inclusion axioms [8]. OWL 2 TBoxes satisfy some syntactic conditions involving the role hierarchy and the appearance of roles in concepts of the form $\exists R.\mathsf{Self}$, $\geq k\, R.C$, $\leq k\, R.C$ and in the assertions $\mathsf{Irr}(P)$, $\mathsf{Dis}(R_1, R_2)$. See [8] for details. The *role depth* of $\mathcal{T}$ is the maximal nesting of constructors involving roles in $\mathcal{T}$.

In DLs, the extensional knowledge about individuals is represented in an ABox. An OWL 2 *ABox*, $\mathcal{A}$, is a finite set of *membership assertions* of the form $C(a)$, $P(a, b)$, and $\neg P(a, b)$. TBox and ABox constitute a *knowledge base* $\langle \mathcal{T}, \mathcal{A} \rangle$.

The semantics of DLs is given in terms of first-order interpretations [3], and the constructs and axioms of OWL 2 are interpreted in the standard way, see [8]. We just mention that for an interpretation $\mathcal{I}$, we have that $(\exists R.\mathsf{Self})^{\mathcal{I}} = \{x \mid (x, x) \in R^{\mathcal{I}}\}$, since this construct is not usually found in DL languages.

OWL 2 is a very expressive DL, but this expressiveness comes at a price. Indeed, reasoning over an OWL 2 ontology is 2ExpTime-hard, and the best known upper bound is 2NExpTime [9]. Also, it is open whether answering conjunctive queries is decidable.

***DL-Lite*$_\mathcal{A}$.** *DL-Lite*$_\mathcal{A}$ has been specifically designed for efficient reasoning and query answering over large amounts of data [15,4]. A *DL-Lite*$_\mathcal{A}$ ontology is formed using atomic concept names A, atomic role names P, and individual names a. In *DL-Lite*$_\mathcal{A}$, we distinguish *basic concepts* B, that may appear in the lhs of concept inclusions, from arbitrary concepts L (for 'light') that may appear only in the rhs of inclusions:

$$B ::= \bot \mid A \mid \exists R \qquad L ::= B \mid \neg B \mid \exists R.L \qquad R ::= P \mid P^-$$

[1] http://www.w3.org/TR/owl2-overview/

[2] For simplicity, we restrict the attention to the features of $\mathcal{SROIQ}$/OWL 2 that are relevant for our purposes.

We call *chain* a sequence $S = R_1 \circ \cdots \circ R_n$ of roles, and use $\exists S$ for $\exists R_1 . \ldots \exists R_n$.

A *DL-Lite*$_\mathcal{A}$ *TBox*, $\mathcal{T}$, is a finite set of: *(i) concept inclusion axioms* $B \sqsubseteq L$, *(ii) role inclusion axioms* $R_1 \sqsubseteq R_2$, and *(iii) role constraints* $\mathsf{Fun}(R)$, $\mathsf{Dis}(R_1, R_2)$, $\mathsf{Asym}(P)$, and $\mathsf{Sym}(P)$, with the syntactic condition that no role that is functional or whose inverse is functional can appear in the rhs of a role inclusion axiom or in a qualified existential restriction [15,4]. Concept inclusions of the form $B \sqsubseteq \exists R_1 . \ldots \exists R_n . B'$ are called *chain inclusions*, and n is the *length* of the chain inclusion.

3 Semantic Approximation from OWL 2 to *DL-Lite*$_\mathcal{A}$

We start by providing the formal definition of the problem we are addressing, that is we define the notion of sound and complete approximation.

Let $\mathcal{L}$ be a description logic. An $\mathcal{L}$ *axiom* is an axiom allowed in $\mathcal{L}$, and an $\mathcal{L}$ *TBox* is a TBox that contains only $\mathcal{L}$ axioms. The *signature* Σ of an $\mathcal{L}$ TBox $\mathcal{T}$ is the alphabet of concept, role, and individual names occurring in $\mathcal{T}$. Let $\mathcal{T}$ and $\mathcal{T}'$ be two $\mathcal{L}$ TBoxes such that $\mathcal{T} \subseteq \mathcal{T}'$. Following [12], we say that $\mathcal{T}'$ is a *conservative extension* of $\mathcal{T}$, if for every axiom I over the signature Σ of $\mathcal{T}$ s.t. $\mathcal{T}' \models I$ we also have that $\mathcal{T} \models I$.

Given two DLs $\mathcal{L}$ and $\mathcal{L}'$, we say that $\mathcal{L}$ is *(syntactically) more expressive* than $\mathcal{L}'$, denoted $\mathcal{L}' \preceq \mathcal{L}$, if every $\mathcal{L}'$ TBox is also an $\mathcal{L}$ TBox.

Definition 1. *Let $\mathcal{L}'$ and $\mathcal{L}$ be two DLs with $\mathcal{L}' \preceq \mathcal{L}$, and let $\mathcal{T}$ be an $\mathcal{L}$ TBox with signature Σ. An* approximation *of $\mathcal{T}$ in $\mathcal{L}'$ is an $\mathcal{L}'$ TBox $\mathcal{T}'$ over a signature $\Sigma' = \Sigma \cup \Sigma_{new}$, where Σ_{new} is a possibly empty set of new names.*

- *$\mathcal{T}'$ is a* sound *approximation (w.r.t. TBox reasoning) if for every $\mathcal{L}'$ axiom I over Σ s.t. $\mathcal{T}' \models I$, we have that $\mathcal{T} \models I$.*
- *$\mathcal{T}'$ is a* complete *approximation (w.r.t. TBox reasoning) if for every $\mathcal{L}'$ axiom I over Σ s.t. $\mathcal{T}' \cup \{I\}$ is an $\mathcal{L}'$ TBox and $\mathcal{T} \models I$, we have that $\mathcal{T}' \models I$.*

The work of [13] allows one to capture in an approximation the basic concept hierarchy entailed by the input ontology that is formulated in an expressive DL such as OWL 2. The approach can be summarized as follows: *(i)* for each pair of basic concepts B_1, B_2 in the signature of the original ontology, check (using a DL reasoner) whether the original ontology implies $B_1 \sqsubseteq B_2$ or $B_1 \sqsubseteq \neg B_2$, *(ii)* for each direct and inverse role R check whether $\mathsf{Fun}(R)$ is implied, and *(iii)* collect all entailments (the so called entailment set) in a *DL-Lite*$_\mathcal{F}$ ontology, i.e., the approximated ontology.

The algorithm in [13] is sound and complete when the target language is *DL-Lite*$_\mathcal{F}$. However, proceeding in this way when the target language is *DL-Lite*$_\mathcal{A}$ will result in incomplete approximations, specifically w.r.t. to entailments involving existential chains. The following example demonstrates this.

Example 2. Consider the TBox $\mathcal{T}_1$ constituted by the axioms: $A \sqsubseteq \exists R_1 . (A_1 \sqcup A_2)$, $A_1 \sqsubseteq \exists R_2 . A$, and $A_2 \sqsubseteq \exists R_2 . A$. One can see that $\mathcal{T}_1$ implies the chain inclusion $A \sqsubseteq \exists R_1 . \exists R_2 . A$, which is not in the entailment set computed as illustrated above. ∎

In *DL-Lite*$_\mathcal{A}$, incompleteness of the approximation is due to missed entailments involving existential chains. Assuring that the approximation entails all possible existential chains is non trivial, since in principle there can be an infinite number of these

entailments. For instance, in Example 2, $\mathcal{T}_1$ implies all chain inclusions of even length of the form $A \sqsubseteq \exists R_1.\exists R_2.\ldots\exists R_1.\exists R_2.A$. Hence, it is clear that the naive approach of enumerating all possible entailments is not viable for *DL-Lite*$_\mathcal{A}$. However, we show that we can preprocess the original ontology by introducing new concepts so that we can resort to checking the existence of chains of a limited length (determined by the role depth in the original ontology), and therefore, limit the number of entailments that we need to check.

Specifically, our algorithm computes the approximation in two steps:

1. We analyze $\mathcal{T}$ to understand which *given* complex concepts can give rise to existential chains. Based on such an analysis, we create an intermediate TBox $\mathcal{T}'$ in which we introduce new named concepts, one for each discovered complex concept. $\mathcal{T}'$ turns out to be a conservative extension of $\mathcal{T}$. In this way, in $\mathcal{T}'$ we can detect all chain inclusions of the form $B \sqsubseteq \exists S.B'$, where S is a chain of limited length, and B, B' are basic concepts in $\mathcal{T}'$.
2. We approximate $\mathcal{T}$ to a *DL-Lite*$_\mathcal{A}$ TBox $\mathcal{T}_A$ by checking all relevant entailments of $\mathcal{T}'$. Due to the extended alphabet we are able to guarantee completeness in a finite number of entailment checks.

We now elaborate on the details of our approach. In Section 3.1, we describe the construction of $\mathcal{T}'$ and in Section 3.2, we provide an algorithm for constructing $\mathcal{T}_A$ and show its correctness.

3.1 Preparation: Introducing New Names

Let $\mathcal{T}$ be an OWL 2 TBox that we want to approximate. We first construct a new OWL 2 TBox $\mathcal{T}'$, which is a conservative extension of $\mathcal{T}$, by introducing named concepts for some of the complex concepts occurring in $\mathcal{T}$. The intuition behind this operation is that giving names to non-*DL-Lite*$_\mathcal{A}$ concepts that qualify existential chains in $\mathcal{T}$ will later allow us to use these 'names' as junctions between several chain inclusions of a restricted length. The difficulty here is to introduce just enough names so that we can guarantee that all existential chains are captured but not more. Thus, we do not give names to the concepts of the form $\exists R_1.\ldots\exists R_n.A$ occurring on the right-hand side of concept inclusions or of the form $A_1 \sqcup A_2$ on the left-hand side of concept inclusions: these are valid *DL-Lite*$_\mathcal{A}$ expressions. However, obviously, we need to name any non *DL-Lite*$_\mathcal{A}$ concept, such as $A_1 \sqcup A_2$ in Example 2.

We define now how to construct $\mathcal{T}'$.

Definition 2. *Let $\mathcal{T}$ be an OWL 2 TBox. Then for every axiom $I \in \mathcal{T}$ we have that $I \in \mathcal{T}'$. Moreover:*

- *if a concept C of the form $\{a_1, \ldots, a_m\}$, $\forall R.C_1$, $\neg C_1$, or $C_1 \sqcup \cdots \sqcup C_m$ appears in $\mathcal{T}$, then add to $\mathcal{T}'$ the concept inclusion axiom $A_C \equiv C$,*
- *if a concept C of the form $C_1 \sqcap \cdots \sqcap C_m$ appears in $\mathcal{T}$ in an inclusion of the form $C' \sqsubseteq \exists S.C$, where S is a chain of roles, then add to $\mathcal{T}'$ the axiom $A_C \equiv C$,*
- *if a concept C of the form $C_1 \sqcap \cdots \sqcap C_m$ or $\exists R_1.\ldots\exists R_m.C'$ appears in $\mathcal{T}$ on the left-hand side of a concept inclusion, then add to $\mathcal{T}'$ the axiom $A_C \equiv C$,*

where A_C is a newly introduced concept name.

The following result is an immediate consequence of the fact that in $\mathcal{T}'$ the newly introduced concept names are asserted to be equivalent to the concepts they stand for.

Lemma 1. *Let $\mathcal{T}$ be an OWL 2 TBox and $\mathcal{T}'$ the TBox obtained from $\mathcal{T}$ according to Definition 2. Then $\mathcal{T}'$ is a conservative extension of $\mathcal{T}$.*

Now, we show how we use $\mathcal{T}'$ for computing approximations. The purpose of extending $\mathcal{T}$ with new names is to be able to restrict the attention to a limited number of entailment checks, specifically checks of chain inclusions. We can detect all chain inclusions implied by $\mathcal{T}$ by looking at chain inclusions of the form $B \sqsubseteq \exists S.B'$ with B, B' basic concepts of $\mathcal{T}'$, and S a chain of roles of length limited by the *role depth* in $\mathcal{T}$. The following result establishes a useful property of the constructed TBox $\mathcal{T}'$.

Proposition 1. *Let $\mathcal{T}$ be an OWL 2 TBox of role depth k, and $\mathcal{T}'$ the TBox obtained from $\mathcal{T}$ according to Definition 2. Let further $\mathcal{T} \models B \sqsubseteq \exists S.B'$, where B, B' are basic concepts of $\mathcal{T}$, and S is a chain of roles of $\mathcal{T}$ (of arbitrary length). Then there are chains $S_1, \ldots, S_m$ of roles of $\mathcal{T}$, all of length at most k, and basic concepts $B_0, \ldots, B_m$ of $\mathcal{T}'$ such that $B_0 = B$, $B_m = B'$, $S = S_1 \circ \cdots \circ S_m$, and $\mathcal{T}' \models B_{i-1} \sqsubseteq \exists S_i.B_i$ for $1 \leq i \leq m$.*

3.2 Constructing a *DL-Lite*$_\mathcal{A}$ TBox

Now, using $\mathcal{T}'$, we show how to construct a *DL-Lite*$_\mathcal{A}$ TBox $\mathcal{T}_A$ that is a sound and complete approximation of $\mathcal{T}$.

Definition 3. *Let $\mathcal{T}$ be an OWL 2 TBox of role depth k, and $\mathcal{T}'$ the TBox obtained from $\mathcal{T}$ according to Definition 2. Then, the DL-Lite$_\mathcal{A}$ approximation of $\mathcal{T}$ is the TBox $\mathcal{T}_A$ constructed as follows. We set $\mathcal{T}_A = \emptyset$ and execute the following sequence of steps:*

1. *for all basic concepts B_1, B_2 of $\mathcal{T}'$, if $\mathcal{T}' \models B_1 \sqsubseteq B_2$, then add $B_1 \sqsubseteq B_2$ to $\mathcal{T}_A$;*
2. *for all basic concepts B_1, B_2 of $\mathcal{T}'$, if $\mathcal{T}' \models B_1 \sqsubseteq \neg B_2$, then add $B_1 \sqsubseteq \neg B_2$ to $\mathcal{T}_A$;*
3. *for all atomic or inverse roles R_1, R_2, if $\mathcal{T}' \models R_1 \sqsubseteq R_2$, then add $R_1 \sqsubseteq R_2$ to $\mathcal{T}_A$;*
4. *for all atomic or inverse roles R_1, R_2 (atomic roles P), if $\mathcal{T}' \models \mathsf{Dis}(R_1, R_2)$ (resp., $\mathsf{Asym}(P)$, $\mathsf{Sym}(P)$), then add $\mathsf{Dis}(R_1, R_2)$ (resp., $\mathsf{Asym}(P)$, $\mathsf{Sym}(P)$) to $\mathcal{T}_A$;*
5. *for all basic concepts B_1, B_2 of $\mathcal{T}'$ and atomic or inverse roles $R_1, \ldots, R_l$, $l < k$, if $\mathcal{T}' \models B_1 \sqsubseteq \exists R_1 . \ldots \exists R_l . B_2$, then add $B_1 \sqsubseteq \exists R_1 . \ldots \exists R_l . B_2$ to $\mathcal{T}_A$;*
6. *for all atomic or inverse roles R, if $\mathcal{T}' \models \mathsf{Fun}(R)$, R does not have proper subroles in $\mathcal{T}_A$, and neither R nor R^- appear in a qualified existential of $\mathcal{T}_A$, then add $\mathsf{Fun}(R)$ to $\mathcal{T}_A$.*

Theorem 1. *Let $\mathcal{T}$ be an OWL 2 TBox and $\mathcal{T}_A$ defined according to Definition 3. Then, $\mathcal{T}_A$ is a sound and complete DL-Lite$_\mathcal{A}$ approximation of $\mathcal{T}$.*

The following result establishes the complexity of computing a sound and complete approximation in *DL-Lite*$_\mathcal{A}$.

Theorem 2. *Let $\mathcal{T}$ be an OWL 2 TBox of role depth k, and Σ the signature of $\mathcal{T}$. Then the algorithm for constructing $\mathcal{T}_A$ according to Definition 3 performs a number of OWL 2 entailment checks that is exponential in k and polynomial in the number of elements of Σ.*

3.3 Cleaning the alphabet: Removing new named concepts

The sound and complete approximation constructed as described above requires to extend the original alphabet. In some situations this might not be desirable, e.g., in those cases where all terms in the ontology need to be 'understandable' by the end-user. However, in order to achieve completeness of the approximation w.r.t. existential chains, such an alphabet extension is in general unavoidable. We believe that a good compromise is a limit on the length of the existential chain entailments captured by the approximation. This limit is reasonable since the presence of existential chains becomes relevant mostly in the context of query answering. We have seen that in this context one can safely assume that the length of the queries will not go beyond a certain limit. Moreover, in cases, in which the ontology is used in a running application, it is reasonable to expect that the queries that are going to be asked to the reasoner are known in advance, as is the case in applications built on top of traditional RDBMS engines. With these observations in mind we can define a limit on the chains based on this length and we will be certain that we are sound and complete in the context of our queries/application.

In order to achieve this, we need to modify the construction of the approximated ontology $\mathcal{T}_A$. Let k be the role depth in $\mathcal{T}$, and ℓ the maximum length of queries. Then, we replace Rule 5 in Definition 3 with the following:

5a. if $\mathcal{T}' \models B \sqsubseteq \exists R_1 . \dots \exists R_{l_1} . A_1$, $\mathcal{T}' \models A_1 \sqsubseteq \exists R_{l_1+1} . \dots \exists R_{l_2} . A_2$, …, $\mathcal{T}' \models A_{m-1} \sqsubseteq \exists R_{l_{m-1}+1} . \dots \exists R_n . B'$, with $m \geq 1$, A_i, for $1 \leq i \leq m-1$ new names in $\mathcal{T}'$, $l_i < k$, and B, B' basic concepts of $\mathcal{T}$, then $B \sqsubseteq \exists R_1 . \dots \exists R_n . B'$ is in $\mathcal{T}_A$;

5b. if $\mathcal{T}' \models B \sqsubseteq \exists R_1 . \dots \exists R_{l_1} . A_1$, $\mathcal{T}' \models A_1 \sqsubseteq \exists R_{l_1+1} . \dots \exists R_{l_2} . A_2$, …, $\mathcal{T}' \models A_{m-1} \sqsubseteq \exists R_{l_{m-1}+1} . \dots \exists R_n . A_m$, with A_i, for $1 \leq i \leq m$ new names in $\mathcal{T}'$, $l_i < k$, $n \leq \ell$, and B a basic concept of $\mathcal{T}$, then $B \sqsubseteq \exists R_1 . \dots \exists R_n$ is in $\mathcal{T}_A$.

Theorem 3. *Let $\mathcal{T}$ be an OWL 2 TBox of role depth k and $\mathcal{T}_A$ the TBox obtained by the Rules 1-4, 5a, 5b, and 6. Then $\mathcal{T}_A$ is a sound and complete approximation of $\mathcal{T}$ in the languages of DL-Lite$_\mathcal{A}$ in which existential chains are limited to the maximum length ℓ.*

4 Implementation

We have implemented the proposed algorithm for *DL-Lite*$_\mathcal{A}$ approximations, as well as a slightly extended version of the algorithm proposed in [13]. The former is a naive, straightforward implementation of the described technique. It is neither optimized w.r.t. run-time nor w.r.t. the size of the output ontology. These implementations are available at `https://babbage.inf.unibz.it/trac/obdapublic/wiki/approx_semantic_index` in three forms: *(i)* a Java API; *(ii)* a command line application suitable for batch approximations; *(iii)* a plug-in for Protégé 4.0. The core algorithm of these modules can work in two modes:

Simple Approximations. This is the algorithm from [13], extended with the ability to capture qualified existential restrictions of length 1 on the right-hand side of concept inclusions. This mode provides sound approximations, which however are incomplete for *DL-Lite*$_\mathcal{A}$ due to the reasons we have explained above;

Complete approximations. This is the algorithm presented in this paper. It is able to construct sound and complete *DL-Lite*$_\mathcal{A}$ approximations with a possibly *extended*

alphabet, or *DL-Lite*$_\mathcal{A}$ approximations in the *original alphabet* that are sound and complete w.r.t. chains of maximum length ℓ. In both modes, we resort to publicly available OWL 2 (or OWL) reasoners, such as Pellet[3], to check for entailments.

Using this implementation we confirmed that indeed even relatively simple ontologies, such as the 'Pizza'[4] ontology, do entail the kind of existential chains that we are interested in. Moreover, as intended, our algorithms are able to capture these chains in practice and we can use our approximations for query answering successfully.

With respect to performance, we found that the exponential nature of the algorithm for complete *DL-Lite*$_\mathcal{A}$ approximations does limit the scope of the usage scenarios in which the technique is applicable. Consider that the approximation of the 'Pizza' ontology took approximately 30 minutes to complete on a windows Vista machine equipped with 2Gb of RAM and an Intel Core 2 Duo processor. In usage scenarios, in which the ontology rarely changes, this performance is acceptable, for example, when the domain ontology is finished and the approximated ontology is only recalculated when there are updates on the former. In contrast, in scenarios where the ontology is dynamic, the high cost of our approximations will be problematic.

With respect to the output ontology, we found that the number of generated assertions is not adequate, especially for those scenarios in which the result should be inspected by humans. For example, in the case of the Pizza ontology we found that an approximation keeping the original alphabet and sound and complete with respect to chains of length 3 generated 130,424 axioms.

5 Conclusions

We have shown that providing sound and complete approximations of OWL 2 to *DL-Lite*$_\mathcal{A}$ ontologies is non-trivial due to the entailment of axioms involving existential chains of unbounded length. We have also provided an algorithm that is able to compute these approximations by locating the sources of these entailments. The core idea of the algorithm is the introduction of new named concepts that allow us to only consider a limited number of chain inclusions. We have also shown that if the approximation is to be complete, in general it is necessary to extend the alphabet. However, if this extended alphabet is not desirable, then it is possible to maintain the original alphabet as long as we put a limit on the length of the entailed chains. A reasonable reference limit is naturally given by the maximum length of queries that are issued over the ontology.

We will focus on further refinements to the proposed techniques. On the one hand, we aim at devising methods to eliminate redundant axioms, generated by the current algorithm. On the other hand we aim at developing methods for the incremental computation of the approximation in dynamic scenarios, to overcome the long processing times that we have currently observed also with 'average case' ontologies such as the ones in public repositories. We are further working on extending our technique towards obtaining (possibly sound and complete) approximations for TBox and ABox reasoning, and for query answering in OWL 2 (See [14] for a first simple solution in this direction).

[3] `http://clarkparsia.com/pellet`

[4] `http://www.co-ode.org/ontologies/pizza/`

References

1. A. Acciarri, D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, M. Palmieri, and R. Rosati. QUONTO: QUerying ONTOlogies. In *Proc. of AAAI 2005*, pages 1670–1671, 2005.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of IJCAI 2005*, pages 364–369, 2005.
3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition, 2007.
4. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, and R. Rosati. Ontologies and databases: The *DL-Lite* approach. In S. Tessaris and E. Franconi, editors, *Semantic Technologies for Informations Systems – 5th Int. Reasoning Web Summer School (RW 2009)*, volume 5689 of *LNCS*, pages 255–356. Springer, 2009.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. of Automated Reasoning*, 39(3):385–429, 2007.
6. B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *J. of Web Semantics*, 6(4):309–322, 2008.
7. S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In R. Meersman, Z. Tari, and S. Stevens, editors, *Database Semantic: Semantic Issues in Multimedia Systems*, chapter 20, pages 351–370. Kluwer Academic Publishers, 1999.
8. I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible $\mathcal{SROIQ}$. In *Proc. of KR 2006*, pages 57–67, 2006.
9. Y. Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. In *Proc. of KR 2008*, pages 274–284, 2008.
10. C. M. Keet, R. Alberts, A. Gerber, and G. Chimamiwa. Enhancing web portals with Ontology-Based Data Access: the case study of South Africa's Accessibility Portal for people with disabilities. In *Proc. of OWLED 2008*, 2008.
11. A. Krisnadhi and C. Lutz. Data complexity in the $\mathcal{EL}$ family of description logics. In *Proc. of LPAR 2007*, pages 333–347, 2007.
12. C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. of IJCAI 2007*, pages 453–458, 2007.
13. J. Z. Pan and E. Thomas. Approximating OWL-DL ontologies. In *Proc. of AAAI 2007*, pages 1434–1439, 2007.
14. J. Z. Pan, E. Thomas, and Y. Zhao. Completeness guaranteed approximations for OWL-DL query answering. In *Proc. of DL 2009*, volume 477 of *CEUR,* `ceur-ws.org`, 2009.
15. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.
16. A. Poggi, M. Rodríguez-Muro, and M. Ruzzi. Ontology-based database access with DIG-Mastro and the OBDA Plugin for Protégé. In K. Clark and P. F. Patel-Schneider, editors, *Proc. of OWLED 2008 DC*, 2008.
17. Y. Ren, J. Z. Pan, and Y. Zhao. Soundness preserving approximation for TBox reasoning. In *Proc. of AAAI 2010*, 2010.
18. B. Selman and H. Kautz. Knowledge compilation and theory approximation. *J. of the ACM*, 43(2):193–224, 1996.
19. T. Tserendorj, S. Rudolph, M. Krötzsch, and P. Hitzler. Approximate OWL-reasoning with Screech. In *Proc. of RR 2008*, volume 5341 of *LNCS*, pages 165–180. Springer, 2008.