Finite model reasoning in *DL-Lite*

Riccardo Rosati

Dipartimento di Informatica e Sistemistica
Sapienza Università di Roma
Via Ariosto 25, 00185 Roma, Italy
rosati@dis.uniroma1.it

Abstract. The semantics of OWL-DL and its subclasses are based on the classical semantics of first-order logic, in which the interpretation domain may be an infinite set. This constitutes a serious expressive limitation for such ontology languages, since, in many real application scenarios for the Semantic Web, the domain of interest is actually finite, although the exact cardinality of the domain is unknown. Hence, in these cases the formal semantics of the OWL-DL ontology does not coincide with its intended semantics. In this paper we start filling this gap, by considering the subclasses of OWL-DL which correspond to the logics of the *DL-Lite* family, and studying reasoning over finite models in such logics. In particular, we mainly consider two reasoning problems: deciding satisfiability of an ontology, and answering unions of conjunctive queries (UCQs) over an ontology. We first consider the description logic *DL-Lite*$_R$ and show that, for the two above mentioned problems, finite model reasoning coincides with classical reasoning, i.e., reasoning over arbitrary, unrestricted models. Then, we analyze the description logics *DL-Lite*$_F$ and *DL-Lite*$_A$. Differently from *DL-Lite*$_R$, in such logics finite model reasoning does not coincide with classical reasoning. To solve satisfiability and query answering over finite models in these logics, we define techniques which reduce polynomially both the above reasoning problems over finite models to the corresponding problem over arbitrary models. Thus, for all the *DL-Lite* languages considered, the good computational properties of satisfiability and query answering under the classical semantics also hold under the finite model semantics. Moreover, we have effectively and easily implemented the above techniques, extending the *DL-Lite* reasoner QuOnto with support for finite model reasoning.

1 Introduction

The semantics of OWL-DL [3] and its fragments [8] are based on the classical semantics of first-order logic, in which the interpretation domain may be either a finite or an infinite set. This constitutes a serious expressive limitation for these ontology languages, since in many real application scenarios for the Semantic Web, the domain of interest is actually finite, although the exact cardinality of the domain is unknown. Hence, in these cases the formal semantics of the OWL-DL ontology does not coincide with its intended semantics.

We illustrate the above problem through two simple examples (in the following examples, the ontologies are expressed using the *DL-Lite*$_F$ language, which corresponds to a fragment of OWL-DL: *DL-Lite*$_F$ is formally introduced in Section 2).

Example 1. Let $\mathcal{O}$ be the following ontology about employees:

$$\mathit{Employee} \sqsubseteq \exists \mathit{isHelpedBy} \tag{1}$$

$$\exists \mathit{isHelpedBy}^{-} \sqsubseteq \mathit{Employee} \tag{2}$$

$$\exists \mathit{isHelpedBy}^{-} \sqsubseteq \mathit{HighSalary} \tag{3}$$

$$(\mathit{funct}\ \mathit{isHelpedBy}^{-}) \tag{4}$$

$$\mathit{Employee}(\mathit{Paul}) \tag{5}$$

which formalizes the following knowledge about the concepts *Employee*, *HighSalary* and the role *isHelpedBy*:

- every employee has some colleague who is committed to help her/him to perform some special task (assertion (1));
- those who are committed to help employees are also employees (assertion (2)) and have a high salary (assertion (3));
- an employee can commit to help at most one of her/his colleagues (assertion (4));
- Paul is an employee (assertion (5)).

Now, on the one hand, it can be shown that the ontology $\mathcal{O}$ does not entail (under the standard OWL-DL semantics) $\mathit{HighSalary}(\mathit{Paul})$: indeed, consider the following interpretation $\mathcal{I}$ over the countably infinite domain $\{d_0, \ldots, d_n, \ldots\}$:

- $\mathit{Paul}^{\mathcal{I}} = d_0$;
- $d_i \in \mathit{Employee}^{\mathcal{I}}$ for each $i \geq 0$;
- $d_i \in \mathit{HighSalary}^{\mathcal{I}}$ for each $i \geq 1$;
- $\langle d_i, d_{i+1} \rangle \in \mathit{isHelpedBy}^{\mathcal{I}}$ for each $i \geq 0$.

It is immediate to verify that $\mathcal{I}$ satisfies the ontology $\mathcal{O}$ and that $\mathit{HighSalary}(\mathit{Paul})$ is not satisfied in $\mathcal{I}$.

On the other hand, *every finite model for $\mathcal{O}$ satisfies $\mathit{HighSalary}(\mathit{Paul})$*. In fact, if the domain is finite, then the chain of employees induced by the ontology on the binary relation *isHelpedBy* must be finite, so the only possible way to close such a chain is to "come back" to the initial employee, i.e., Paul, who is the only employee who does not help someone yet. Consequently, in every finite model for $\mathcal{O}$, Paul helps some colleague, hence he has a high salary.

Now, for the above ontology $\mathcal{O}$ it seems very natural to assume that the domain of interest is always finite (although not exactly known), i.e., it is unreasonable to assume as possible the existence of an infinite number of employees. Hence, in this case we would like to conclude from the above ontology that Paul has a high salary. However, all current OWL reasoners will not derive such a conclusion. ■

Example 2. Let $\mathcal{O}$ be the following ontology about peer networks:

$$\mathit{EUpeer} \sqsubseteq \exists \mathit{hasNAmirror} \tag{6}$$

$$\exists \mathit{hasNAmirror}^{-} \sqsubseteq \mathit{NApeer} \tag{7}$$

$$\mathit{NApeer} \sqsubseteq \exists \mathit{hasEUmirror} \tag{8}$$

$$\exists \mathit{hasEUmirror}^- \sqsubseteq \mathit{EUpeer} \tag{9}$$

$$\exists \mathit{hasEUmirror}^- \sqsubseteq \mathit{AlwaysOnline} \tag{10}$$

$$(\mathit{funct}\ \mathit{hasNAmirror}^-) \tag{11}$$

$$(\mathit{funct}\ \mathit{hasEUmirror}^-) \tag{12}$$

$$\mathit{EUpeer}(\texttt{dis.uniroma1.it}) \tag{13}$$

which formalizes the following knowledge about the concepts *EUpeer*, *NApeer*, *AlwaysOnline* and the roles *hasNAmirror*, *hasEUmirror*:

- every European peer has a mirror who is a North American peer (assertions (6) and (7));
- every North American peer has a mirror who is a European peer (assertions (8) and (9));
- peers who are mirrors of North American peers are always on-line (assertion (10));
- a peer can be the mirror of at most one North American peer (assertion (11)) and of at most one European peer (assertion (12));
- `dis.uniroma1.it` is a European peer (assertion (13)).

In a way similar to the previous example, it can be shown that $\mathcal{O}$ does not entail (under the standard OWL-DL semantics) $\mathit{AlwaysOnline}(\texttt{dis.uniroma1.it})$, while every finite model for $\mathcal{O}$ satisfies $\mathit{AlwaysOnline}(\texttt{dis.uniroma1.it})$. Similarly to the previous example, also for the above ontology $\mathcal{O}$ it seems very natural to assume that the domain of interest is always finite. Hence, we would like to conclude that peer `dis.uniroma1.it` is always on-line, but all current OWL reasoners will not derive such a conclusion. ■

The above examples highlight the fact that OWL-DL ontologies (under the classical first-order semantics) lack the ability to properly handle finite domains of interest. In this respect, it is also important to point out that database applications are always based on a finite domain assumption: such an assumption should be taken into account in ontology-based access to database information sources, which is going to be one of the prominent applications of the Semantic Web technology [12].

Finite model reasoning has actually been studied in past research in Description Logics (DLs) [2], which constitute the logical basis of OWL-DL. In particular, besides previous studies summarized in [2], recent research has mainly focused on $\mathcal{ALCQI}$, a large fragment of OWL-DL [4, 10]: for such a logic, an EXPTIME-completeness result for finite model reasoning has been established [10]. However, such results on finite model reasoning in DLs only consider "classical" DL reasoning tasks (concept subsumption, knowledge base satisfiability) and do not take *query answering* into account. Moreover, none of the currently available (OWL)DL reasoners supports finite model reasoning.

Thus, with respect to finite model reasoning in OWL-DL ontologies, we are still missing: (i) a thorough computational analysis, in particular for tractable fragments of OWL-DL [8]; (ii) an analysis of query answering over finite models; (iii) the implementation of ontology-management systems supporting finite model reasoning.

In this paper we start filling this gap, by considering the subclasses of OWL-DL which correspond to the logics of the *DL-Lite* family of Description Logics [6], and study reasoning over finite models in such logics. In particular, we consider three reasoning problems: deciding entailment of intensional assertions (TBox entailment), deciding satisfiability of the ontology, and answering queries (specifically, unions of conjunctive queries) over an ontology.

Our contributions can be summarized as follows:

1. We first consider the description logic *DL-Lite*$_R$ and show that, for all the above mentioned problems, finite model reasoning coincides with classical reasoning, i.e., reasoning over arbitrary, unrestricted models.
2. Then, we analyze the description logics *DL-Lite*$_F$ and *DL-Lite*$_A$. Differently from *DL-Lite*$_R$, in such logics finite model reasoning does not coincide with classical reasoning. To solve TBox entailment, satisfiability and query answering over finite models in these logics, we define techniques which reduce polynomially all the above reasoning problems over finite models in such logics to the corresponding problem over arbitrary models. This allows us to show that, for all the *DL-Lite* languages considered, the good computational properties of TBox entailment, satisfiability and query answering under the classical semantics also hold under the finite semantics.
3. Finally, we have effectively and easily implemented the above techniques to provide the *DL-Lite* reasoner QuOnto with support for finite model reasoning. To the best of our knowledge, such an extension of QuOnto constitutes the first ontology management system providing an automated support to finite model reasoning.

2 The description logics *DL-Lite*$_F$ and *DL-Lite*$_R$

2.1 Syntax

We start from three mutually disjoint alphabets: an alphabet of concept names, an alphabet of role names, and an alphabet of constant (or individual) names. We call *basic concept* an expression of the form $B ::= A \mid \exists P \mid \exists P^-$, where A is a concept name and P is a role name, and we call *basic role* an expression of the form $R ::= P \mid P^-$, where P is a role name.

A *DL-Lite*$_R$ TBox assertion is an expression of one of the following forms:

- $B_1 \sqsubseteq B_2$ (concept inclusion assertion) where B_1, B_2 are basic concepts;
- $R_1 \sqsubseteq R_2$ (role inclusion assertion) where R_1, R_2 are basic roles;
- $B_1 \sqsubseteq \neg B_2$ (concept disjointness assertion) where B_1, B_2 are basic concepts;
- $R_1 \sqsubseteq \neg R_2$ (role disjointness assertion) where R_1, R_2 are basic roles;

A *DL-Lite*$_F$ TBox assertion is an expression of one of the following forms:

- $B_1 \sqsubseteq B_2$ (concept inclusion assertion) where B_1, B_2 are basic concepts;
- $B_1 \sqsubseteq \neg B_2$ (concept disjointness assertion) where B_1, B_2 are basic concepts;
- $(funct\ R)$ (functionality assertion) where R is a basic role.

A *DL-Lite*$_R$ TBox is a set of *DL-Lite*$_R$ TBox assertions, while a *DL-Lite*$_F$ TBox is a set of *DL-Lite*$_F$ TBox assertions.

A *membership assertion* is a ground atom, i.e., an expression of the form $A(a)$, $P(a, b)$ where A is a concept name, P is a role name, and a, b are constant names.

An ABox is a set of membership assertions.

A *DL-Lite*$_R$ *knowledge base* (KB) is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is a *DL-Lite*$_R$ TBox and $\mathcal{A}$ is an ABox. Analogously, a *DL-Lite*$_F$ KB is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is a *DL-Lite*$_F$ TBox and $\mathcal{A}$ is an ABox.

We call *DL-Lite*$_F^-$ TBox a *DL-Lite*$_F$ TBox without disjointness assertions. Analogously, We call *DL-Lite*$_R^-$ TBox a *DL-Lite*$_R$ TBox without disjointness assertions.

We now introduce queries. A *union of conjunctive query (UCQ)* is an expression of the form

$$\{x_1, \ldots, x_n \mid \mathit{conj}_1 \vee \ldots \vee \mathit{conj}_m\}$$

where $n \geq 0$, $m \geq 1$, and each *conj*$_i$ is an expression of the form $(\exists y_1, \ldots, y_j . a_1 \wedge \ldots \wedge a_h)$, where $j \geq 0$, $h \geq 1$, each a_i is an atom, i.e., an expression of the form $A(t)$ or $P(t, t')$ where A is a concept name, P is a role name and t, t' are either constants or variables from $\{x_1, \ldots, x_n, y_1, \ldots, y_j\}$. When $n = 0$ (i.e., the above expression is a first-order sentence) the UCQ is called a *Boolean* UCQ.

2.2 Semantics

The semantics of a DL is given in terms of interpretations, where an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty *interpretation domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ that assigns to each concept C a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and to each role R a binary relation $R^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$. In particular, we have:

$$\begin{aligned}
A^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \\
P^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \\
(P^-)^{\mathcal{I}} &= \{(o_2, o_1) \mid (o_1, o_2) \in P^{\mathcal{I}}\} \\
(\exists R)^{\mathcal{I}} &= \{o \mid \exists o' . (o, o') \in R^{\mathcal{I}}\} \\
(\neg B)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus B^{\mathcal{I}} \\
(\neg R)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}
\end{aligned}$$

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is called *finite* if $\Delta^{\mathcal{I}}$ is a finite set.

An interpretation $\mathcal{I}$ is a *model* of $B \sqsubseteq C$, where C is either a basic concept or the negation of a basic concept, if $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$. Similarly, $\mathcal{I}$ is a *model* of $R \sqsubseteq E$, where R is a basic role and E is either a basic role or the negation of a basic role, if $R^{\mathcal{I}} \subseteq E^{\mathcal{I}}$.

To specify the semantics of membership assertions, we extend the interpretation function to constants, by assigning to each constant a a *distinct* object $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Note that this implies that we enforce the *unique name assumption* on constants [2]. An interpretation $\mathcal{I}$ is a model of a membership assertion $A(a)$, (resp., $P(a, b)$) if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (resp., $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$).

Given an (inclusion, or membership) assertion ϕ, and an interpretation $\mathcal{I}$, we denote by $\mathcal{I} \models \phi$ the fact that $\mathcal{I}$ is a model of ϕ. Given a (finite) set of assertions Φ, we denote by $\mathcal{I} \models \Phi$ the fact that $\mathcal{I}$ is a model of every assertion in Φ. A *model of a KB* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

is an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. A finite interpretation that is a model of a KB $\mathcal{K}$ is called *finite model* of $\mathcal{K}$.

The reasoning tasks we are interested in are UCQ entailment, KB satisfiability, and TBox entailment. More precisely:

- A KB $\mathcal{K}$ *entails* a Boolean UCQ q, denoted by $\mathcal{K} \models q$, if all models of $\mathcal{K}$ are also models of q (where an interpretation $\mathcal{I}$ is a model of q if $\mathcal{I}$ satisfies the first-order sentence q according to the standard notion of satisfiability in first-order logic). $\mathcal{K}$ finitely entails a UCQ q, denoted by $\mathcal{K} \models_{\mathit{fin}} q$, if all finite models of $\mathcal{K}$ are also models of q;
- a KB is *satisfiable* if it has at least one model, and is *finitely satisfiable* if it has at least one finite model.
- a TBox $\mathcal{T}$ *entails* a TBox assertion ϕ, denoted by $\mathcal{T} \models \phi$, if all models of $\mathcal{T}$ are also models of ϕ, while $\mathcal{T}$ *finitely entails* ϕ, denoted by $\mathcal{T} \models_{\mathit{fin}} \phi$, if all finite models of $\mathcal{T}$ are also models of ϕ.

Since in the following we will focus on UCQ entailment, from now on when we speak about UCQs we always mean *Boolean* UCQs. We recall that answering arbitrary (i.e., non-Boolean) UCQs can be easily reduced to UCQ entailment (for more details see e.g. [9]).

3 Finite model reasoning in *DL-Lite*$_R$

In this section we show that, in *DL-Lite*$_R$, finite model reasoning coincides with classical reasoning. We start by proving such property for *DL-Lite*$_R^-$ KBs (i.e., *DL-Lite*$_R$ KBs without disjointness assertions) and for UCQ entailment.

Lemma 1. *Let $\mathcal{T}$ be a DL-Lite$_R^-$-TBox, and let q be a UCQ. Then, for every ABox $\mathcal{A}$, $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathit{fin}} q$ iff $\langle \mathcal{T}, \mathcal{A} \rangle \models q$.*

Proof. The proof is a consequence of Theorem 4 of [11], since a *DL-Lite*$_R$ KB corresponds to a database with inclusion dependencies (interpreted under open-world assumption). More precisely, given a *DL-Lite*$_R$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, we build a database instance (i.e., a set of facts) $\mathcal{D}$ and a set of inclusion dependencies $\mathcal{C}$ as follows:

- for every concept name A (respectively, role name R) occurring in $\mathcal{K}$, the database schema contains a unary relation A (respectively, a binary relation R)
- for every concept inclusion in $\mathcal{T}$ of the form $A_1 \sqsubseteq A_2$, $\mathcal{C}$ contains the inclusion dependency $A_1[1] \subseteq A_2[1]$;
- for every concept inclusion in $\mathcal{T}$ of the form $A \sqsubseteq \exists R$ (respectively, $A \sqsubseteq \exists R^-$), $\mathcal{C}$ contains the inclusion dependency $A[1] \subseteq R[1]$ (respectively, $A[1] \subseteq R[2]$);
- for every concept inclusion in $\mathcal{T}$ of the form $\exists R \sqsubseteq A$ (respectively, $\exists R^- \sqsubseteq A$), $\mathcal{C}$ contains the inclusion dependency $R[1] \subseteq A[1]$ (respectively, $R[2] \subseteq A[1]$);
- for every concept inclusion in $\mathcal{T}$ of the form $\exists R_1 \sqsubseteq \exists R_2$ (respectively, $\exists R_1 \sqsubseteq \exists R_2^-$), $\mathcal{C}$ contains the inclusion dependency $R_1[1] \subseteq R_2[1]$ (respectively, $R_1[1] \subseteq R_2[2]$);

- for every concept inclusion in $\mathcal{T}$ of the form $\exists R_1^- \sqsubseteq \exists R_2$ (respectively, $\exists R_1^- \sqsubseteq \exists R_2^-$), $\mathcal{C}$ contains the inclusion dependency $R_1[2] \subseteq R_2[1]$ (respectively, $R_1[2] \subseteq R_2[2]$);
- for every role inclusion in $\mathcal{T}$ of the form $R_1 \sqsubseteq R_2$ (respectively, $R_1 \sqsubseteq R_2^-$), $\mathcal{C}$ contains the inclusion dependency $R_1[1,2] \subseteq R_2[1,2]$ (respectively, $R_1[1,2] \subseteq R_2[2,1]$);
- for every role inclusion in $\mathcal{T}$ of the form $R_1^- \sqsubseteq R_2$ (respectively, $R_1^- \sqsubseteq R_2^-$), $\mathcal{C}$ contains the inclusion dependency $R_1[2,1] \subseteq R_2[1,2]$ (respectively, $R_1[2,1] \subseteq R_2[2,1]$).

Finally, for every membership assertion of the form $A(a)$ (respectively, $R(a,b)$) in the ABox $\mathcal{A}$, the database instance $\mathcal{D}$ contains the fact $A(a)$ (respectively, $R(a,b)$).

We now recall the definition of [11] of semantics of the pair $(\mathcal{C},\mathcal{D})$: we denote by *sem*$(\mathcal{C},\mathcal{D})$ the set of database instances $\{\mathcal{B} \mid \mathcal{B} \supseteq \mathcal{D}$ and $\mathcal{B}$ satisfies $\mathcal{C}\}$, where each $\mathcal{B}$ is a (possibly infinite) set of facts. Moreover, we denote by *sem*$_f(\mathcal{C},\mathcal{D})$ the subset of *sem*$(\mathcal{C},\mathcal{D})$ where each $\mathcal{B}$ is a *finite* set of facts.

It is now immediate to verify that the set of models of $\mathcal{K}$ is in one-to-one correspondence with *sem*$(\mathcal{C},\mathcal{D})$: more precisely, every database instance $\mathcal{B}$ in *sem*$(\mathcal{C},\mathcal{D})$ corresponds to a model $\mathcal{I}(\mathcal{B})$ of $\mathcal{K}$ where $\mathcal{I}(\mathcal{B})$ is the interpretation defined as follows: for each concept name C and for each constant a, $a \in C^{\mathcal{I}(\mathcal{B})}$ iff $C(a) \in \mathcal{B}$, and for each role name R and pair of constant a, b, $\langle a,b\rangle \in R^{\mathcal{I}(\mathcal{B})}$ iff $R(a,b) \in \mathcal{B}$.

This in turn immediately implies that for every UCQ q, $\langle \mathcal{T},\mathcal{A}\rangle \models q$ iff q is true in all database instances of *sem*$(\mathcal{C},\mathcal{D})$, and $\langle \mathcal{T},\mathcal{A}\rangle \models_{\mathit{fin}} q$ iff q is true in all database instances of *sem*$_f(\mathcal{C},\mathcal{D})$. Since from Theorem 4 of [11] q is true in all database instances of *sem*$(\mathcal{C},\mathcal{D})$ iff q is true in all database instances of *sem*$_f(\mathcal{C},\mathcal{D})$, the thesis follows. □

We now extend the above result to arbitrary *DL-Lite*$_R$ KBs (i.e., KBs whose TBox may also contain disjointness assertions).

Given a *DL-Lite*$_R$ TBox $\mathcal{T}$, let $D(\mathcal{T})$ denote the set of disjointness assertions occurring in $\mathcal{T}$, and let $\mathcal{T}^- = \mathcal{T} - D(\mathcal{T})$, i.e., $\mathcal{T}^-$ denotes the TBox obtained from $\mathcal{T}$ by eliminating all disjointness assertions.

Given a disjointness assertion ϕ, we denote by $Q(\phi)$ the Boolean conjunctive query defined as follows:

- if $\phi = A_1 \sqsubseteq \neg A_2$, then $Q(\phi) = \exists x.A_1(x) \wedge A_2(x)$;
- if $\phi = A \sqsubseteq \neg\exists R$ or $\phi = \exists R \sqsubseteq \neg A$, then $Q(\phi) = \exists x,y.A(x) \wedge R(x,y)$;
- if $\phi = A \sqsubseteq \neg\exists R^-$ or $\phi = \exists R^- \sqsubseteq \neg A$, then $Q(\phi) = \exists x,y.A(x) \wedge R(y,x)$;
- if $\phi = \exists R_1 \sqsubseteq \neg\exists R_2$, then $Q(\phi) = \exists x,y,z.R_1(x,y) \wedge R_2(x,z)$;
- if $\phi = \exists R_1 \sqsubseteq \neg\exists R_2^-$ or $\phi = \exists R_2^- \sqsubseteq \neg\exists R_1$, then $Q(\phi) = \exists x,y,z.R_1(x,y) \wedge R_2(z,x)$;
- if $\phi = \exists R_1^- \sqsubseteq \neg\exists R_2^-$, then $Q(\phi) = \exists x,y,z.R_1(y,x) \wedge R_2(z,x)$.

Informally, $Q(\phi)$ is satisfied in an interpretation $\mathcal{I}$ iff $\mathcal{I}$ does not satisfy the disjointness assertion ϕ.

Furthermore, given a set of disjointness assertions Φ, we define $Q(\Phi)$ as the following UCQ:

$$Q(\Phi) = \bigvee_{\phi \in \Phi} Q(\phi)$$

Lemma 2. *Let $\mathcal{T}$ be a DL-Lite$_R$-TBox, and let q be a UCQ. Then, for every UCQ q, $\langle \mathcal{T}, \mathcal{A} \rangle \models q$ iff $\langle \mathcal{T}^-, \mathcal{A} \rangle \models q \vee Q(D(\mathcal{T}))$ and $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathit{fin}} q$ iff $\langle \mathcal{T}^-, \mathcal{A} \rangle \models_{\mathit{fin}} q \vee Q(D(\mathcal{T}))$.*

Proof. It is immediate to verify that $D(\mathcal{T})$ corresponds to a first-order sentence ϕ which is equivalent to $\neg Q(D(\mathcal{T}))$, therefore from the deduction theorem it follows that $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathit{fin}} q$ iff $\langle \mathcal{T}^-, \mathcal{A} \rangle \models_{\mathit{fin}} q \vee Q(D(\mathcal{T}))$. Moreover, since $\mathcal{T}^-$ is a *DL-Lite*$_R^-$ TBox, from Theorem 1 it follows that $\langle \mathcal{T}^-, \mathcal{A} \rangle \models_{\mathit{fin}} q \vee Q(D(\mathcal{T}))$ iff $\langle \mathcal{T}^-, \mathcal{A} \rangle \models q \vee Q(D(\mathcal{T}))$. □

As an immediate consequence of Lemma 2 and of Lemma 1, we obtain the following property.

Theorem 1. *Let $\mathcal{T}$ be a DL-Lite$_R$-TBox, and let q be a UCQ. Then, for every ABox $\mathcal{A}$, $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathit{fin}} q$ iff $\langle \mathcal{T}, \mathcal{A} \rangle \models q$.*

Then, we turn our attention to KB satisfiability. It can easily be shown that the technique for reducing KB unsatisfiability in *DL-Lite*$_R$ to UCQ entailment (see Lemma 16 of [6]) is also correct when we restrict to finite models. This fact and Theorem 1 imply the following property.

Theorem 2. *Let $\mathcal{K}$ be a DL-Lite$_R$ KB. Then, $\mathcal{K}$ is finitely satisfiable iff $\mathcal{K}$ is satisfiable.*

Moreover, it can also be shown that the technique for reducing TBox entailment in *DL-Lite*$_R$ to KB unsatisfiability (see Theorem 22 and Theorem 23 of [6]) is also correct when we restrict to finite models. Consequently, the two previous theorems imply the following property.

Theorem 3. *Let $\mathcal{T}$ be a DL-Lite$_R$-TBox, and let ϕ be a DL-Lite$_R$ TBox assertion. Then, $\mathcal{T} \models_{\mathit{fin}} \phi$ iff $\mathcal{T} \models \phi$.*

4 Finite model reasoning in *DL-Lite*$_F$

In this section we study finite model reasoning in *DL-Lite*$_F$. First, we remark that, differently from the case of *DL-Lite*$_R$, in *DL-Lite*$_F$ UCQ entailment over finite models differs from UCQ entailment over arbitrary models, as illustrated by both Example 1 and Example 2. So, since we cannot simply establish an equivalence between classical reasoning and finite model reasoning as in the case of *DL-Lite*$_R$, we must look for new reasoning methods to solve finite model reasoning in *DL-Lite*$_F$.

We start by considering *DL-Lite*$_F^-$ KBs (i.e., *DL-Lite*$_F$ KBs without disjointness assertions) and define inference rules for TBox assertions in *DL-Lite*$_F^-$.

4.1 Finite TBox inference rules in *DL-Lite*$_F^-$

In the following, R denotes a basic role expression (i.e., either P or P^- where P is a role name), while R^- denotes the inverse of R, i.e., if $R = P$ (R is a role name), then $R^- = P^-$, while if $R = P^-$, then $R^- = P$.

Definition 1. *Given a DL-Lite$_F^-$ TBox $\mathcal{T}$, $\mathit{finClosure}_F(\mathcal{T})$ denotes the closure of $\mathcal{T}$ with respect to the following inference rules:*

1. *(*inclusion-rule*) if $B_1 \sqsubseteq B_2$ and $B_2 \sqsubseteq B_3$ then conclude $B_1 \sqsubseteq B_3$;*
2. *(*functionality-inclusion-cycle-rule*) if there is an inclusion-functionality cycle, i.e., a sequence of TBox assertions of the form*

$$(\mathit{funct}\ R_1),\ \exists R_2 \sqsubseteq \exists R_1^-,\ (\mathit{funct}\ R_2),\ \exists R_3 \sqsubseteq \exists R_2^-,\ \ldots$$
$$\ldots,\ (\mathit{funct}\ R_k),\ \exists R_1 \sqsubseteq \exists R_k^-$$

(where each R_i is a basic role expression, i.e., either P or P^-), then conclude

$$(\mathit{funct}\ R_1^-),\ \exists R_1^- \sqsubseteq \exists R_2,\ (\mathit{funct}\ R_2^-),\ \exists R_2^- \sqsubseteq \exists R_3, \ldots$$
$$\ldots,\ (\mathit{funct}\ R_k^-),\ \exists R_k^- \sqsubseteq \exists R_1$$

It is immediate to verify that the above inference rules are not sound with respect to classical TBox entailment (i.e., entailment over unrestricted models). On the other hand, we now prove that the above inference rules are sound with respect to TBox entailment over finite models.

Lemma 3. *Let $\mathcal{T}$ be a DL-Lite$_F^-$ TBox and let ϕ be a DL-Lite$_F^-$ TBox assertion. If $\mathit{finClosure}_F(\mathcal{T}) \models \phi$ then $\mathcal{T} \models_{\mathit{fin}} \phi$.*

Proof. The proof is a consequence of the axiomatization for unary inclusion dependencies and functional dependencies presented in [7]. □

We call a TBox *f-closed* if it is closed with respect to the two inference rules above.

4.2 Query answering over finite models in *DL-Lite$_F^-$*

We now turn our attention to query answering over finite models in *DL-Lite$_F^-$* KBs. Notice that Lemma 3 does not imply that query answering over finite models can be solved by simply augmenting the TBox with the new assertions implied by Definition 1. However, we now show that this strategy is actually complete, i.e., we can answer (unions of) conjunctive queries over finite models of a *DL-Lite$_F^-$* KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ by first generating the augmented TBox $\mathcal{T}'$ obtained from $\mathcal{T}$ by adding the new assertions implied by Definition 1, and then computing the certain answers to queries over the KB $\langle \mathcal{T}', \mathcal{A} \rangle$ according to the unrestricted semantics.

Theorem 4. *Let $\mathcal{T}$ be a DL-Lite$_F^-$-TBox, and let q be a UCQ. Then, for every ABox $\mathcal{A}$, $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathit{fin}} q$ iff $\langle \mathit{finClosure}_F(\mathcal{T}), \mathcal{A} \rangle \models q$.*

Proof (sketch). The proof is rather involved and is based on the notion of *chase* of a *DL-Lite* KB [6]. More precisely, we modify the notion of (generally infinite) chase of a *DL-Lite$_F$* KB, denoted by $\mathit{chase}(\mathcal{K})$ and presented in [6], thus defining the notion of *finite* chase of degree k of $\mathcal{K}$, denoted by $\mathit{finChase}_k^F(\mathcal{K})$.

Informally, $\mathit{chase}(\mathcal{K})$ is an ABox built starting from the initial ABox $\mathcal{A}$ and applying a chase expansion rule that adds new membership assertions to the ABox until all inclusions in $\mathcal{T}$ are satisfied. In this expansion process new constants are introduced in the ABox, and in the presence of cyclic concept inclusions this process may produce an infinite ABox. The important semantic properties of $chase(\mathcal{K})$ are the following: (i) $chase(\mathcal{K})$ is isomorphic to an interpretation $\mathcal{I}(chase(\mathcal{K}))$ which is a model of $\mathcal{K}$; (ii) UCQ entailment over $\mathcal{K}$ can be solved by simply evaluating the query over $\mathcal{I}(chase(\mathcal{K}))$.

We modify the above chase procedure and define a chase procedure, called $\mathit{finChase}_k^F(\mathcal{K})$ (parameterized with respect to a positive integer k), that always terminates, thus returning a finite ABox. Then, we prove that $\mathit{finChase}_k^F(\mathcal{K})$ only partially preserves the above semantic properties of the chase with respect to finite model semantics. More precisely, we prove that:

(A) $\mathit{finChase}_k^F(\mathcal{K})$ is isomorphic to an interpretation $\mathcal{I}(\mathit{finChase}_k^F(\mathcal{K}))$ which is a model of $\mathcal{K}$, *under the assumption that the TBox $\mathcal{T}$ of $\mathcal{K}$ is f-closed*;

(B) for every positive integer k, entailment of a UCQ *of length less or equal to k* (i.e., all of whose conjunctions have a number of atoms less or equal to k) can be decided by simply evaluating the query over $\mathcal{I}(\mathit{finChase}_k^F(\mathcal{K}))$.

With the notion of $\mathit{finChase}_k^F(\mathcal{K})$ in place, we can prove the theorem as follows. First, let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and $\mathcal{K}' = \langle \mathit{finClosure}_F(\mathcal{T}), \mathcal{A} \rangle$. Soundness is trivial: if $\mathcal{K} \models_{\mathit{fin}} q$, then $\mathcal{K}' \models_{\mathit{fin}} q$, and since the set of finite models of $\mathcal{K}'$ is a subset of the set of models of $\mathcal{K}'$, it follows that $\mathcal{K}' \models q$. To prove completeness, assume that $\mathcal{K}' \not\models q$ and let k be the length of q. Now, by property (A) above, $\mathcal{I}(\mathit{finChase}_k^F(\mathcal{K}'))$ is a finite model of $\mathcal{K}'$, and by Lemma 3, $\mathcal{I}(\mathit{finChase}_k^F(\mathcal{K}'))$ is a model of $\mathcal{K}$. Finally, since $\mathcal{K}' \not\models q$, from the above property (B) it follows that $\mathcal{I}(\mathit{finChase}_k^F(\mathcal{K}')) \not\models q$, which in turn implies that $\mathcal{K} \not\models_{\mathit{fin}} q$. □

4.3 Query answering over finite models in *DL-Lite*$_F$

It is now immediate to extend the above theorem to *DL-Lite*$_F$, since it is possible to encode disjointness assertions in the UCQ, as illustrated already in Section 3. Given a *DL-Lite*$_F$ TBox $\mathcal{T}$, let $D(\mathcal{T})$ denote the set of disjointness assertions occurring in $\mathcal{T}$, and let $\mathcal{T}^- = \mathcal{T} - D(\mathcal{T})$, i.e., $\mathcal{T}^-$ denotes the *DL-Lite*$_F^-$ TBox obtained from $\mathcal{T}$ by eliminating all disjointness assertions.

Theorem 5. *Let $\mathcal{T}$ be a DL-Lite$_F$-TBox, and let q be a UCQ. Then, for every ABox $\mathcal{A}$, $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathit{fin}} q$ iff $\langle \mathit{finClosure}_F(\mathcal{T}^-), \mathcal{A} \rangle \models q \vee Q(D(\mathcal{T}))$.*

4.4 Finite KB satisfiability and TBox entailment

We now focus on KB satisfiability. Again, we start by showing that, differently from the case of *DL-Lite*$_R$, in *DL-Lite*$_F$ finite KB satisfiability does not coincide with classical KB saisfiability. In particular, there are *DL-Lite*$_F$ KBs that only admit infinite models.

Example 3. Let $\mathcal{T}$ be the following *DL-Lite*$_F$ TBox:

$$A \sqsubseteq \exists R \qquad A \sqsubseteq \neg \exists R^- \qquad \exists R^- \sqsubseteq \exists R \qquad (\mathit{funct}\ R^-)$$

and let $\mathcal{A}$ be the ABox $\mathcal{A} = \{A(a)\}$. It is easy to see that the KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is not finitely satisfiable, while $\mathcal{K}$ is satisfiable (i.e., there are models for $\mathcal{K}$ but they are all infinite). ■

To compute finite KB satisfiability, it is possible to show that the technique for reducing KB unsatisfiability in *DL-Lite*$_F$ to UCQ entailment (see Lemma 16 of [6]) is also correct when we restrict to finite models. This fact and Theorem 5 imply that we can reduce finite KB satisfiability in *DL-Lite*$_F$ to standard KB satisfiability. Formally, the following property holds:

Theorem 6. *Let $\mathcal{K}$ be a DL-Lite$_F$ KB. Then, $\mathcal{K}$ is finitely satisfiable iff $\langle \mathcal{T} \cup \mathit{finClosure}_F(\mathcal{T}^-), \mathcal{A} \rangle$ is satisfiable.*

It also turns out that finite TBox entailment in *DL-Lite*$_F$ can be reduced to finite KB unsatisfiability. In fact, it is easy to show that Theorem 22 and Theorem 24 of [6] hold also when restricting to finite models, while Theorem 25 of [6] holds for finite models under the assumption that the TBox $\mathcal{T}$ is f-closed. Consequently, the following property holds.

Theorem 7. *Let $\mathcal{T}$ be a DL-Lite$_F$-TBox, and let ϕ be a DL-Lite$_F$ TBox assertion. Then, $\mathcal{T} \models_{\mathit{fin}} \phi$ iff $\mathcal{T} \cup \mathit{finClosure}_F(\mathcal{T}^-) \models \phi$.*

5 Complexity results

We now study the computational complexity of finite model reasoning in *DL-Lite*$_R$ and *DL-Lite*$_F$.

First, in the case of *DL-Lite*$_R$, the theorems shown in Section 3 immediately imply that, for the reasoning tasks considered in this paper, the complexity of finite model reasoning and of classical reasoning coincide. Hence, the complexity results for *DL-Lite*$_R$ reported in [6] also holds when restricting to finite models.

We now analyze complexity in the case of *DL-Lite*$_F$. First, we show the following property.

Lemma 4. *Given a DL-Lite$_F^-$ TBox $\mathcal{T}$ and a DL-Lite$_F^-$ TBox assertion ϕ, $\mathcal{T} \models_{\mathit{fin}} \phi$ can be decided in polynomial time with respect to the size of $\mathcal{T} \cup \{\phi\}$.*

Moreover, by definition, $\mathit{finClosure}_F(\mathcal{T})$ is only composed of *DL-Lite*$_F^-$ TBox assertions using concept and role names occurring in $\mathcal{T}$, and the number of possible *DL-Lite*$_F^-$ TBox assertions using such names is quadratic with respect to the size of $\mathcal{T}$. Thus, from the above lemma, it follows that $\mathit{finClosure}_F(\mathcal{T})$ can be computed in polynomial time with respect to the size of $\mathcal{T}$. Furthermore, it is immediate to see that $Q(D(\mathcal{T}))$ can also be computed in polynomial time with respect to the size of $\mathcal{T}$.

Therefore, from the theorems shown in Section 4, it follows that the complexity results for *DL-Lite*$_F$ reported in [6] also holds when restricting to finite models.

The above results are formally summarized as follows:

Theorem 8. *Deciding UCQ entailment over finite models in DL-Lite$_R$ and DL-Lite$_F$ is:*

- *in LOGSPACE with respect to the size of the ABox;*
- *in PTIME with respect to the size of the KB;*
- *NP-complete with respect to the size of the KB and the query.*

Theorem 9. *Let $\mathcal{K}$ be either a DL-Lite$_R$ or a DL-Lite$_F$ KB. Deciding whether $\mathcal{K}$ is finitely satisfiable is:*

- *in LOGSPACE with respect to the size of the ABox;*
- *in PTIME with respect to the size of the TBox;*
- *in PTIME with respect to the size of the KB.*

Theorem 10. *Finite entailment of an assertion ϕ with respect to a TBox $\mathcal{T}$ in DL-Lite$_R$ and DL-Lite$_F$ can be decided in PTIME with respect to the size of $\mathcal{T} \cup \{\phi\}$.*

6 Extension to *DL-Lite$_A$*

In this section we extend the previous results for finite model reasoning to the case of *DL-Lite$_A$*. Due to space limitations, in the present version of the paper we are not able to introduce *DL-Lite$_A$* in detail (see [5]) and just sketch the way in which we have extended the previous results to the case of *DL-Lite$_A$* KBs.

Informally, a *DL-Lite$_A$* TBox is a TBox which admits all the TBox assertions allowed in both *DL-Lite$_F$* and *DL-Lite$_R$* with the following limitation: a functional role cannot be specialized, i.e., if the assertion $(\mathit{funct}\ R)$ or $(\mathit{funct}\ R^-)$ is in the TBox, then there is no role inclusion assertion of the form $R' \sqsubseteq R$ or of the form $R' \sqsubseteq R^-$ in the TBox. Moreover, *DL-Lite$_A$* allows for defining concept *attributes*, i.e, binary relations whose ranges, called value-domains, are concepts interpreted over a domain that is disjoint from the interpretation domain of ordinary concept and roles. A *DL-Lite$_A$* TBox allows for value-domain inclusion/disjointness assertions and for attribute inclusion/disjointness assertions. We refer the reader to [5] for more details.

First, we denote by *DL-Lite$_A^-$* the version of *DL-Lite$_A$* that does not allow for (concept or role or value-domain or attribute) disjointness assertions.

Given a *DL-Lite$_A^-$* TBox $\mathcal{T}$, we denote by $\mathit{finClosure}_A(\mathcal{T})$ the closure of $\mathcal{T}$ with respect to the following inference rules:

1. (*binary-inclusion-rule*) if $R_1 \sqsubseteq R_2$ is either a role inclusion assertion or an attribute inclusion assertion, then conclude $\exists R_1 \sqsubseteq \exists R_2$ and $\exists R_1^- \sqsubseteq \exists R_2^-$;
2. (*transitivity-rule*) if $B_1 \sqsubseteq B_2$ and $B_2 \sqsubseteq B_3$ then conclude $B_1 \sqsubseteq B_3$;
3. (*functionality-inclusion-cycle-rule*) if there is an inclusion-functionality cycle, i.e., a sequence of TBox assertions of the form

$$\begin{array}{l}(\mathit{funct}\ R_1),\ \ \exists R_2 \sqsubseteq \exists R_1^-,\ \ (\mathit{funct}\ R_2),\ \ \exists R_3 \sqsubseteq \exists R_2^-,\ \ \ldots\\ \ldots,\ \ (\mathit{funct}\ R_k),\ \ \exists R_1 \sqsubseteq \exists R_k^-\end{array}$$

(where each R_i is a basic role expression, i.e., either P or P^-), then conclude

$$\begin{array}{l}(\mathit{funct}\ R_1^-),\ \ \exists R_1^- \sqsubseteq \exists R_2,\ \ (\mathit{funct}\ R_2^-),\ \ \exists R_2^- \sqsubseteq \exists R_3,\ \ \ldots\\ \ldots,\ \ (\mathit{funct}\ R_k^-),\ \ \exists R_k^- \sqsubseteq \exists R_1\end{array}$$

Observe that, with respect to $\mathit{finClosure}_F$, to compute $\mathit{finClosure}_A$ we basically just add inference rules for deriving unary (i.e., concept and value-domain) inclusions from binary (i.e., role and attribute) inclusions.

We are now able to prove that the reduction of query answering over finite models to query answering over unrestricted models proved in Section 4 for *DL-Lite*$_F$ can be extended to *DL-Lite*$_A$.

Theorem 11. *Let $\mathcal{T}$ be a DL-Lite$_A$-TBox, and let q be a UCQ. Then, for every ABox $\mathcal{A}$, $\langle \mathcal{T}, \mathcal{A} \rangle \models_{\mathit{fin}} q$ iff $\langle \mathit{finClosure}_A(\mathcal{T}^-), \mathcal{A} \rangle \models q \vee Q(D(\mathcal{T}))$.*

Proof. First, consider a *DL-Lite*$_A^-$ TBox $\mathcal{T}$. Let $U(\mathcal{T})$ denote the set of unary (i.e., concept and value-domain) inclusions and functionality assertions in $\mathcal{T}$.

We first modify the procedure for computing $\mathit{finChase}_k^F$, illustrated in the proof of Theorem 4, thus producing a new terminating chase procedure $\mathit{finChase}_k^A$. Then, we prove that: (A) $\mathit{finChase}_k^A(\mathcal{K})$ is isomorphic to an interpretation $\mathcal{I}(\mathit{finChase}_k^A(\mathcal{K}))$ that is a model of $\mathcal{K}$; (B) for every positive integer k, entailment of a UCQ of length less or equal to k can be decided by simply evaluating the query over $\mathcal{I}(\mathit{finChase}_k^A(\mathcal{K}))$. From the above two properties, the thesis follows with a proof analogous to the proof of Theorem 4.

Finally, let us consider the case when $\mathcal{T}$ is an arbitrary *DL-Lite*$_A$ TBox, i.e., when $\mathcal{T}$ also contains disjointness assertions. In this case, the thesis is proved in the same way as in the proof of Theorem 5. □

From the above theorem, and in the same way as in the case of *DL-Lite*$_F$, we are able to derive the following properties.

Theorem 12. *Let $\mathcal{K}$ be a DL-Lite$_A$ KB. Then, $\mathcal{K}$ is finitely satisfiable iff $\langle \mathcal{T} \cup \mathit{finClosure}_A(\mathcal{T}^-), \mathcal{A} \rangle$ is satisfiable.*

Theorem 13. *Let $\mathcal{T}$ be a DL-Lite$_A$-TBox, and let ϕ be a DL-Lite$_A$ TBox assertion. Then, $\mathcal{T} \models_{\mathit{fin}} \phi$ iff $\mathcal{T} \cup \mathit{finClosure}_A(\mathcal{T}^-) \models \phi$.*

Finally, from the above results it follows that the computational properties expressed by Theorem 8, Theorem 9, and Theorem 10 extend to the case of *DL-Lite*$_A$.

7 Implementation

In this section we show that the techniques presented in this paper allow for an efficient and effective implementation of finite model reasoning services for *DL-Lite* ontologies.

We have implemented the above techniques in QuOnto [1]. QuOnto is a *DL-Lite*$_A$ reasoner whose main purpose is to deal with very large instances (i.e., ABoxes). To this aim, in QuOnto the ABox is stored and managed by a relational database system (RDBMS). The main reasoning services provided by QuOnto are the “extensional” ones, i.e., KB satisfiability and query answering (it allows for posing UCQs over the ontology).

Computation in QuOnto is divided into an off-line and an on-line phase. Off-line processing concerns purely intensional tasks, i.e., upload of the TBox, classification of concept and roles, etc. On-line processing concerns the tasks involving the ABox, i.e., KB satisfiability and query answering. In particular, query answering is divided into a query rewriting phase, in which the query is reformulated in terms of an SQL query (through a reasoning step which exploits the knowledge expressed by the TBox), and a query evaluation phase, in which the generated SQL query is evaluated over the ABox by the RDBMS, and the answers are presented to the user.

To provide support for finite model reasoning, we have extended QuOnto as follows:

- during off-line processing, the system computes $\mathit{finClosure}_A(\mathcal{T})$, and the assertions in $\mathit{finClosure}_A(\mathcal{T}) - \mathcal{T}$ are stored in an auxiliary TBox $\mathcal{T}'$;
- during on-line processing, at every service request (KB satisfiability or query answering) the user may choose between the classical semantics and the finite model semantics;
- if finite model semantics is selected, then the system executes its reasoning service method (KB satisfiability or query answering) using the TBox $\mathcal{T} \cup \mathcal{T}'$, otherwise it executes its method on the TBox $\mathcal{T}$.

Notice that the main additional computation requested is the computation of $\mathit{finClosure}_A(\mathcal{T})$, which is only executed during off-line processing. Hence, finite model reasoning does not imply any significant overhead during on-line processing. Moreover, even the off-line overhead caused by the computation of $\mathit{finClosure}_A(\mathcal{T})$ is usually not very significant, since such a computation is in the worst case quadratic in the number of functional roles of $\mathcal{T}$ (which are usually only a small fraction of the total number of roles in $\mathcal{T}$). This very nice computational behaviour has been confirmed by our experiments. We have thus included these functionalities for finite model reasoning in the next version of QuOnto, which is going to be publicly released in 2008.

8 Conclusions

Comparison with related work We remark that *DL-Lite*$_R$ and *DL-Lite*$_F$ are fragments of expressive Description Logics for which finite model reasoning has been studied in the past [4, 2, 10]. In particular, decidability of finite KB satisfiability and of finite TBox entailment for both *DL-Lite*$_F$ and *DL-Lite*$_R$ is a consequence of such previous results. However, the complexity of such tasks in these two logics was not known, while the PTIME upper bound for finite TBox entailment in *DL-Lite*$_F^-$ is implied by the results in [7]. Furthermore, nothing was known about decidability and complexity of answering conjunctive queries and unions of conjunctive queries over finite models in such logics.

Future work From the theoretical viewpoint, our aim is to extend the present computational analysis towards other species of OWL-DL. In particular, we are interested in the tractable Description Logics which could become part of the standardization process of OWL 1.1 [8]. From a both theoretical and practical viewpoint, we would like to explore the idea of extending the ontology specification language with constructs that allow the user to associate single subparts of the ontology with either finite or unrestricted interpretation.

Acknowledgments The author wishes to thank Maurizio Lenzerini for inspiring the present work. The author also warmly thanks the anonymous reviewers for their comments, in particular for pointing out the relationship between Lemma 3 and the results in [7]. This research has been partially supported by FET project TONES (Thinking ONtologiES), funded by the EU under contract number FP6-7603, by project HYPER, funded by IBM through a Shared University Research (SUR) Award grant, and by MIUR FIRB 2005 project "Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet" (TOCAI.IT).

References

1. QuOnto web site. `http://www.dis.uniroma1.it/~quonto`.
2. 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, 2003.
3. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language reference. W3C Recommendation, Feb. 2004. Available at `http://www.w3.org/TR/owl-ref/`.
4. D. Calvanese. Finite model reasoning in description logics. In *Proc. of KR'96*, pages 292–303, 1996.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati. MASTRO-I: Efficient integration of relational data through dl ontologies. In *Proceedings of the 2007 International Workshop on Description Logic (DL 2007)*. CEUR Electronic Workshop Proceedings, 2007.
6. 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*, 39:385–429, 2007.
7. S. S. Cosmadakis, P. C. Kanellakis, and M. Vardi. Polynomial-time implication problems for unary inclusion dependencies. *J. of the ACM*, 37(1):15–46, Jan. 1990.
8. B. Cuenca Grau. Tractable fragments of the OWL 1.1 Web Ontology Language. `http://www.cs.man.ac.uk/~bcg/tractable.html`.
9. B. Glimm, I. Horrocks, C. Lutz, and U. Sattler. Conjunctive query answering for the description logic SHIQ. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007)*, pages 399–404, 2007.
10. C. Lutz, U. Sattler, and L. Tendera. The complexity of finite model reasoning in description logics. *Information and Computation*, 199:132–171, 2005.
11. R. Rosati. On the decidability and finite controllability of query processing in databases with incomplete information. In *Proc. of PODS 2006*, pages 356–365, 2006.
12. N. Shadbolt, W. Hall, and T. Berners-Lee. The semantic web revisited. *IEEE Intelligent Systems*, May-June 2006:96–101.