Paraconsistent Query Answering Over *DL-Lite* Ontologies

Liping Zhou, Houkuan Huang
Beijing Jiaotong University, Beijing, China
dearliping@gmail.com, hkhuang@bjtu.edu.cn

Guilin Qi
AIFB, Universität Karlsruhe, Germany
gqi@aifb.uni-karlsruhe.de

Yue Ma
LIPN, Université Paris-Nord, France
Yue.Ma@lipn.univ-paris13.fr

Zhisheng Huang
Department of Computer Science,
Vrije Universiteit Amsterdam, The Netherlands
huang@cs.vu.nl

Youli Qu
Beijing Jiaotong University, Beijing, China
ylqu@bjtu.edu.cn

Abstract

Consistent query answering over description logic-based ontologies is an important topic in ontology engineering as it can provide meaningful answers to queries posed over inconsistent ontologies. Current approaches for dealing with this problem usually consist of two steps: the first step is extracting some consistent sub-ontologies of an inconsistent ontology, then posing the query over these sub-ontologies. In this paper, we propose an alternative approach for consistent query answering over *DL-Lite* ontologies based on four-valued semantics, where *DL-Lite* is a family of tractable DLs. We give an algorithm to compute answers to a query over inconsistent *DL-Lite* ontologies and show that it is tractable. In particular, it is PTime in the size of TBox, and LOGSPACE in the size of ABox.

1 Introduction

As a shared conceptualization of a particular domain, ontologies play an important role for the success of the Semantic Web [3]. Description Logics (DLs) are considered as important formal description languages for specifying ontologies. They provide logical underpinning of Web Ontology Language (OWL). For example, the expressive DL $\mathcal{SHOIN}(D)$ underpins OWL DL, which is the key language of OWL. However, expressive DLs suffer from worst-case exponential time behavior of reasoning [2] which may hinder practical applications to very large real life ontologies. As an important tractable DL family, *DL-Lite* can keep all the reasoning tasks with polynomial time complexity in the size of the ontology [5].

As the size of ontologies grows and applications become more complex, inconsistencies frequently occur within the ontology lifecycle, such as ontology construction, ontology evolution and ontology merging. However, conclusions drawn from an inconsistent ontology by classical inference may be completely meaningless according the fact *ex contradictione quodlibet* [6].

For consistent query answering over inconsistent databases or inconsistent propositional knowledge bases, some approaches are known [11, 1, 4, 7]. The approaches in [1, 4] mainly rely on the notion of repair for a database instance that may violate integrity constraints specified over its schema. And also consistent query answering of conjunctive queries expressed over database schemas with integrity constraints is a coNP-complete problem in data complexity [1]. Villadsen [11] proposed a many-valued paraconsistent logic for query answering based on a simple notion of indeterminacy. However he only focus on symbolic logic and mainly discuss instance checking. Lembo et al. [7] proposed an approach for consistent instance checking based on an inconsistency-tolerant semantics. Their method relies on the notion of *repair* by deleting some inconsistent membership assertions, which will lead to the information loss. The above approaches for dealing with the problem of consistent query answering over inconsistent ontologies usually consist of two steps: the first step is extracting some consistent sub-ontologies of an inconsistent ontology, then posing the query over these sub-ontologies. Lembo et al. [7] have shown that their approach to consistent conjunctive query answering over inconsistent *DL-Lite* ontologies is general intractable w.r.t. data complexity. Since *DL-Lite* is a tractable DLs which is suitable for dealing with large amount of data, finding a tractable approach to consistent query answering is an important problem for *DL-Lite*.

In this paper, we propose a tractable approach to consistent query answering over *DL-Lite* ontologies based on four-valued semantics [10, 8]. We first give a four-valued semantics for *DL-Lite*, and then, by extending the notion of chase in [5], we give a notion of *4-chase* based on four-valued semantics. *4-chase* can be used for constructing a four-valued model for *DL-Lite*. Finally, we present an algorithm for consistent query answering over *DL-Lite* whose computational complexity is polynomial with respect to the size of the TBox.

The rest of the paper is organized as follows: Section 2 presents some basic notions for *DL-Lite*. Section 3 gives a four-valued semantics for *DL-Lite*. Section 4 presents an algorithm to compute a four-valued model and to compute the certain answers to a query over an inconsistent *DL-Lite* ontology. We conclude our paper in Section 5.

2 Preliminaries

DL-Lite is a family of Description Logics (DLs) used to capture some of the most popular conceptual modeling formalisms, such as Entity-Relationship model and UML class diagrams, while preserving the tractability of the most important reasoning tasks, such as ontology satisfiability. We mainly consider two important DLs in *DL-Lite* family: *DL-Lite*$_{\mathcal{F}}$ and *DL-Lite*$_{\mathcal{R}}$[5].

The language of *DL-Lite*$_{core}$ is the core language for *DL-Lite*$_{\mathcal{F}}$ and *DL-Lite*$_{\mathcal{R}}$, in which concepts and roles are formed according to the following syntax:

$B \longrightarrow A \mid \exists R \qquad R \longrightarrow P \mid P^{-} \qquad C \longrightarrow B \mid \neg B \qquad E \longrightarrow R \mid \neg R$

where A and P denote an atomic concept and an atomic role respectively; B denotes a *basic concept* (i.e., a concept of the form A, $\exists R$); R denotes a *basic role* (i.e., a role of the form P, P^{-}), where P^{-} denotes the inverse of the atomic role; C denotes a *general concept* (i.e., a concept of the form B, $\neg B$), whereas E denotes a *general role* (i.e., a concept of the form R, $\neg R$).

A *DL-Lite* ontology consists of a TBox and an ABox. A *DL-Lite*$_{core}$ TBox is a set of inclusion axioms of the form $B \sqsubseteq C$. A *DL-Lite*$_{core}$ ABox is a set of membership assertions on atomic concepts and atomic roles: $A(a), P(a, b)$, where a and b are constants.

DL-Lite$_{\mathcal{R}}$ extends *DL-Lite*$_{core}$ with the ability of specifying inclusion assertions between roles of the form $R \sqsubseteq E$, where R and E are defined as above. *DL-Lite*$_{\mathcal{F}}$ extends *DL-Lite*$_{core}$ with the ability of specifying functionality on roles or on their inverses. Assertions used for this purpose are of the form (funct R). Hereinafter, we use the term *DL-Lite* to refer to either *DL-Lite*$_{\mathcal{R}}$ or *DL-Lite*$_{\mathcal{F}}$, we call positive inclusions(PIs) assertions of the form $B_1 \sqsubseteq B_2$ or of the form $R_1 \sqsubseteq R_2$, whereas we call negative inclusions(NIs) assertions of the form $B_1 \sqsubseteq \neg B_2$ or $R_1 \sqsubseteq \neg R_2$.

The semantics of *DL-Lite* is defined via a model-theoretic semantics, which explicates the relationship between the language syntax and the model of a domain: An interpretation $I = (\Delta^I, \cdot^I)$, consisting of a non-empty *interpretation domain* Δ^I and an *interpretation function* $\cdot^I$, which maps from individuals, concepts and roles to elements of the domain, subsets of the domain and binary relations on the domain, respectively. Given an interpretation I, we say that I satisfies a concept inclusion axiom $B \sqsubseteq C$ (resp., a role inclusion axiom $R \sqsubseteq S$) if $B^I \subseteq C^I$ (resp., $R^I \subseteq S^I$). I satisfies a function assertion (funct R) if $\forall d, e, e^{'}, (d, e) \in R^I \wedge (d, e^{'}) \in R^I \rightarrow e = e^{'}$. Furthermore, I satisfies an atomic concept assertion $A(a)$ (resp., an atomic role assertion $P(a, b)$) if $a^I \in A^I$ (resp., $(a^I, b^I) \in P^I$). An interpretation I is called a model of an ontology O, iff it satisfies each axiom and each assertion in O. An ontology is satisfiable if it has at least one model. An ontology $\mathcal{K}$ logically implies an assertion α, written $\mathcal{K} \models \alpha$, if all models of $\mathcal{K}$ are also models of α. The unique name assumption on constants [2] is adapted by *DL-Lite*. Furthermore, *DL-Lite*$_\mathcal{R}$ has the finite model property, that is, if a *DL-Lite*$_\mathcal{R}$ is consistent, then it has a classical model whose domain is finite [2, 5]. However *DL-Lite*$_\mathcal{F}$ does not have finite model property [5].

Calvanese et al. [5] have given a novel interpretation about $\mathcal{A}$ denoted as $db(\mathcal{A}) = \langle \Delta^{db(\mathcal{A})}, \cdot^{db(\mathcal{A})} \rangle$ which is defined as follows:

- $\Delta^{db(\mathcal{A})}$ is the nonempty set consisting of all constants occurring in $\mathcal{A}$;
- $a^{db(\mathcal{A})} = a$, for each constant a;
- $A^{db(\mathcal{A})} = \{a \mid A(a) \in \mathcal{A}\}$, for each atomic concept A;
- $P^{db(\mathcal{A})} = \{(a_1, a_2) \mid P(a_1, a_2) \in \mathcal{A}\}$, for each atomic role P.

A *union of conjunctive queries (UCQ)* q over a *DL-Lite* ontology $\mathcal{K}$ is an expression of the form $q(\vec{x}) \leftarrow \bigvee\limits_{i=1,\cdots n} \exists \vec{y_i}.conj_i(\vec{x}, \vec{y_i})$, where each $conj_i(\vec{x}, \vec{y_i})$ is a conjunction of atoms and equalities, with free variables $\vec{x}$ and $\vec{y_i}$ [5, 7]. Variables in $\vec{x}$ are called distinguished, and the size of $\vec{x}$ is called the *arity* of q. Atoms in each $conj_i$ are of the form $A(z)$ or $P(z_1, z_2)$, where A and P are respectively an atomic concept and an atomic role of $\mathcal{K}$, and z, z_1, z_2 are either constants in $\mathcal{K}$ or variables. A Boolean UCQ is a query with arity 0, written simply as a sentence of the form $\bigvee_{i=1,\cdots n} \exists \vec{y_i}.conj_i(\vec{y_i})$. A UCQ with a single conjunction of atoms is called *conjunctive query* (CQ). Let q be a Boolean UCQ over a *DL-Lite* ontology $\mathcal{K}$. We say that q is entailed by $\mathcal{K}$, and write $\mathcal{K} \models q$, if, for every model $\mathcal{M}$ of $\mathcal{K}$, $\mathcal{M} \models q$. Let q be a non-Boolean UCQ of arity n over $\mathcal{K}$, and let $\vec{t}$ be an n-tuple of constants. We say that $\vec{t}$ is a *certain answer* to q in $\mathcal{K}$ if $\mathcal{K} \models q(\vec{t})$, where $q(\vec{t})$ is the sentence obtained from the body of q by replacing its distinguished variables by constants in $\vec{t}$. We denote by $Ans(q, \mathcal{K})$ the set of certain answers to q in $\mathcal{K}$.

In order to compute $Ans(q, \mathcal{K})$, Calvanese et al. [5] have proposed an algorithm which is used to compute the perfect reformulation of a conjunctive query q, called PerfectRef, which takes as input a UCQ q and a *DL-Lite* TBox $\mathcal{T}$. Roughly speaking, in PerfectRef, all positive inclusions in $\mathcal{T}$ are used as rewriting rules, iteratively applied from right to left to atoms occurring in the query, thus allowing for compiling away in the rewriting the intensional knowledge of $\mathcal{T}$ that is relevant for answering q. We refer the reader to [5] for further details instead of giving here the exact definition of the algorithm PerfectRef. Let us see an example.

Example 1. *Given a DL-Lite ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ =\{PhDStud $\sqsubseteq$ Stud, PhDStud $\sqsubseteq$ Employee, Stud $\sqsubseteq \neg$Employee\}, $\mathcal{A}$ = \{PhDStud(a)\}. Consider a query $q(x) = Stud(x)$. Now, let us see the Execution of the algorithm* PerfectRef*. Because the atom $Stud(x)$ can be applied to the PI PhDStud $\sqsubseteq$ Stud, we obtain a new query $q(x) = PhDStud(x)$. So the result returned by the algorithm is the union of the following query $\{Stud(x), PhDStud(x)\}$.*

After query rewriting by *DL-Lite* reasoners [5], query answering over *DL-Lite* ontologies can be carried out by an SQL engine, so as to take advantage of existing query optimization strategies and algorithms provided by modern database management systems.

3 Four-valued Semantics for *DL-Lite*

For a given domain Δ and a concept A (resp., R), a four-valued interpretation over Δ assigns to A (resp., R) an extended truth value $\langle A_P, A_N \rangle$ (resp., $\langle R_P, R_N \rangle$), where A_P is the subset of Δ (resp., R_P is the

Table 1: Four-valued semantics of *DL-Lite*

Constructor	Semantics
A	$A^I = \langle A_P, A_N\rangle$,where $A_P, A_N \subseteq \Delta^I$
R	$R^I = \langle R_P, R_N\rangle$,where $R_P, R_N \subseteq \Delta^I \times \Delta^I$
R^-	$(R^-)^I = \langle R_P^-, R_N^-\rangle$,where R_P^-, R_N^- represent the inverse relations on R_P and R_N,respectively.
$\neg A$	$(\neg A)^I = \langle A_N, A_P\rangle$
$\neg R$	$(\neg R)^I = \langle R_N, R_P\rangle$
$\exists R$	$(\exists R)^I = \langle \{x \mid \exists y \in \Delta^I, (x,y) \in R_P^I\}, \{x \mid \forall y \in \Delta^I, (x,y) \in R_N^I\}\rangle$
$\neg\exists R$	$(\neg\exists R)^I = \langle \{x \mid \forall y \in \Delta^I, (x,y) \in R_N^I\}, \{x \mid \exists y \in \Delta^I, (x,y) \in R_P^I\}\rangle$
$=$	$(=)^I = \langle =_P, =_N\rangle$, where $=_P, =_N \in \Delta^I \times \Delta^I$

subset of $\Delta\times\Delta$) that supports A (resp., R) to be true and A_N is the subset of Δ (resp., R_N is the subset of $\Delta\times\Delta$) that supports A (resp., R) to be false. We denote $proj^+(\langle P, N\rangle) = P$ and $proj^-(\langle P, N\rangle) = N$ [8]. The four-valued semantics of *DL-Lite* is given by means of an interpretation $I = (\Delta^I, \cdot^I)$ consisting of a non-empty *interpretation domain* Δ^I and an *interpretation function* $\cdot^I$ satisfying the conditions [9] in Table 1. In Table 1, we introduce four-valued semantics to "=" to represent the four-valued semantics of a functionality assertion. For any given domain Δ, we assign to "=" an extended truth value $\langle =_P, =_N\rangle$, where "$=_P$" stands for the set of pairs of constants which are equal and "$=_N$" stands for the set of pairs of constants which are not equal. The UNA can be expressed as $\forall x, y \in \Delta^{db(\mathcal{A})}, (x,y) \in proj^-((=)^I)$.

Based on the four-valued semantics, there are four truth values for membership assertions. The four truth values are $true$, $false$, $contradictory$ and $unkown$, we use the symbols $t, f, \mathbb{B}, \mathbb{N}$ to denote them respectively [8]. The corresponding four-valued semantics for concept assertions is given as follows:

Definition 1. *[8] For any given instance $a \in \Delta^I$ and concept name A,*

- $A^I(a) = t$, *iff* $a \in proj^+(A^I)$ *and* $a \notin proj^-(A^I)$;
- $A^I(a) = f$, *iff* $a \notin proj^+(A^I)$ *and* $a \in proj^-(A^I)$;
- $A^I(a) = \mathbb{B}$, *iff* $a \in proj^+(A^I)$ *and* $a \in proj^-(A^I)$;
- $A^I(a) = \mathbb{N}$, *iff* $a \notin proj^+(A^I)$ *and* $a \notin proj^-(A^I)$.

The corresponding four-valued semantics for role assertion (or equality "=") can be defined in a similar way.

Given a four-valued interpretation I, we say that I satisfies a concept inclusion axiom $B \sqsubseteq C$ (resp., a role inclusion axiom $R_1 \sqsubseteq R_2$) if $proj^+(B^I) \subseteq proj^+(C^I)$ (resp., $proj^+(R_1^I) \subseteq proj^+(R_2^I)$). I satisfies a function assertion (funct P) if $\forall x, y, z, (x,y) \in proj^+(P^I) \wedge (x,z) \in proj^+(P^I) \rightarrow (y,z) \in proj^+((=)^I)$. Furthermore, I satisfies an atomic concept assertion $A(a)$ (resp., an atomic role assertion $P(a,b)$) if $a^I \in proj^+(A^I)$ (resp., $(a^I, b^I) \in proj^+(P^I)$). A four-valued model of a *DL-Lite* ontology $\mathcal{K}$ is a four-valued interpretation I which satisfies each assertion and each axiom in $\mathcal{K}$. A *DL-Lite* ontology is four-valued satisfiable (unsatisfiable) if there exists (does not exist) such a model.

Example 2. *[12] Given a DL-Lite ontology $\mathcal{K} = \langle\mathcal{T}, \mathcal{A}\rangle$, where $\mathcal{T}$ ={PhDStud $\sqsubseteq$ Stud, PhDStud $\sqsubseteq$ Employee, Stud $\sqsubseteq \neg$Employee, Stud $\sqsubseteq \exists$hasTutor,* (funct *hasTutor*)}*,$\mathcal{A}$ = {PhDStud(a), hasTutor(a,b),* *hasTutor$(a,c)\}$. We can find that it is an inconsistent ontology. Consider the following four-valued interpretation $I = (\Delta^I, .^I)$, where $\Delta^I = \{a,b,c\}$, PhDStud$^I = \langle\{a\},\emptyset\rangle$, Stud$^I = \langle\{a\},\emptyset\rangle$, Employee$^I = \langle\{a\},\{a\}\rangle$, hasTutor$^I = \langle\{(a,c),(a,b)\},\emptyset\rangle$, $(=)^I = \langle\{(a,a),(b,b),(c,c),(b,c)\},\{(a,b),(a,c),(b,c)\}\rangle$. We can find that I is a four-valued model of $\mathcal{K}$ and PhDStud$^I(a) = t$, Stud$^I(a) = t$, Employee$^I(a) = \mathbb{B}$ and $(=)^I(b,c) = \mathbb{B}$. It is easy to obtain four-valued semantics for other atomic assertions.*

4 Query answering over *DL-Lite* based on four-valued semantics

In this section, we mainly discuss how to do paraconsistent query answering over an inconsistent *DL-Lite* ontology. Ma et al. [9] have proposed a method for reasoning with inconsistent *DL-Lite* ontologies through reducing an inconsistent ontology under four-valued semantics to a consistent one under classical semantics. One may think that we can also apply the reduction method to an inconsistent ontology and then do query answering over the resulting ontology. However, this solution has some limitations. When transforming an inconsistent ontology to a consistent one, we may need to introduce some new axioms. Therefore, the size of resulting ontology may be much bigger than that of original ontology, this is not desirable for very large real life ontologies. Thus, we adapt the method for query answering in *DL-Lite* by considering four-valued semantics and propose a polynomial time algorithm for answering unions of conjunctive queries over inconsistent *DL-Lite* ontologies.

4.1 Four-valued canonical interpretation

Let us first see the answers or the certain answers to a query over a *DL-Lite* ontology under four-valued semantics which will be used in the following properties.

Definition 2 (*4-Ans*)**.** *Let $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ be a DL-Lite ontology, I be a four-valued model of $\mathcal{K}$, and $q(\vec{x})$ be a CQ $\exists\vec{y}.conj(\vec{x},\vec{y})$ or UCQ $\bigvee_{i=1,\dots,n}\exists y_i.conj_i(\vec{x},\vec{y_i})$. The answer to $q(\vec{x})$ over I, denoted q^I, is the set of tuples $\vec{t}$ of constants of $\mathcal{A}$ such that the formula $\exists\vec{y}.conj(\vec{t},\vec{y})$ or $\bigvee_{i=1,\dots,n}\exists y_i.conj_i(\vec{t},\vec{y_i})$ evaluates to t or $\mathbb{B}$ in I. The certain answer to $q(\vec{x})$, denoted 4-Ans$(q,\mathcal{K})$, is the set of tuples $\vec{t}$ of constants of $\mathcal{A}$ such that $\vec{t} \in q^I$, for every four-valued model I of $\mathcal{K}$.*

Example 3 (Example 2 contd.)**.** *Let us consider a query $q(x) =$ PhDStud$(x) \wedge \exists y.$hasTutor(x,y), we can obtain $q^I = \{a\}$ and 4-Ans$(q,\mathcal{K}) = \{a\}$.*

Notice that by Definition 2, *4-Ans*$(q,\mathcal{K})$ is finite because $\mathcal{K}$ is finite, and hence the number of constants appearing in $\mathcal{A}$ is finite. We also notice that the tuple $\vec{t}$ can be empty tuple in the case when q is a Boolean query. More precisely, in this case the set *4-Ans*$(q,\mathcal{K})$ consists of the only empty tuple if and only is the query q is t or $\mathbb{B}$ in every four-valued model of $\mathcal{K}$.

Under the four-valued semantics, we will use a special membership assertions of the form $\neg A(a)$ or $\neg R(a,b)$ which is not supported in *DL-Lite* under the classical semantics. In the following, we call *positive membership assertions (PMAs)* of the form $A(a)$ or of the form $P(a,b)$, whereas we call *negative membership assertions (NMAs)* of the form $\neg A(a), \neg R(a,b)$. For simplicity, we use the term "membership assertions" to refer to PMAs or NMAs. We will use the symbol "$*$" to denote all constants in the domain Δ. For example, assume a domain $\Delta = \{a,b,c\}$, then $R(a,*)$ denotes the set $\{R(a,a), R(a,b), R(a,c)\}$. For easy illustration, we also use the function ga [5] that takes as input a basic role and two constants and returns a membership assertion, that is, if $R = P$, then $ga(R,a,b) = P(a,b)$; if $R = P^-$, then $ga(R,a,b) = P(b,a)$.

First, we need to extend some definitions in [5]. We start by defining the notion of applicable inclusion or functionality. Then we give a definition of *4-chase*$(\mathcal{K})$ by extending the notion of chase [5] which is defined based on classical semantics. The four-valued canonical interpretation of a *DL-Lite* ontology is a four-valued interpretation constructed according to *4-chase*$(\mathcal{K})$.

Definition 3 (Applicable Inclusion or Functionality)**.** *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite ontology. Suppose $\mathcal{S}$ is a set of membership assertions. An inclusion assertion or a functionality assertion $\alpha \in \mathcal{T}$ is applicable in $\mathcal{S}$ to a membership assertion $f \in \mathcal{S}$ if*
• $\alpha = A_1 \sqsubseteq A_2, f = A_1(a)$ *and* $A_2(a) \notin \mathcal{S}$*;*
• $\alpha = A \sqsubseteq \exists R, f = A(a)$ *and there does not exist any constant* b *such that* $ga(R, a, b) \in \mathcal{S}$*;*
• $\alpha = \exists R \sqsubseteq A, f = ga(R, a, b)$ *and* $A(a) \notin \mathcal{S}$*;*
• $\alpha = \exists R_1 \sqsubseteq \exists R_2, f = ga(R_1, a, b)$ *and there does not exist any constant* c *such that* $ga(R_2,a,c) \in \mathcal{S}$*;*
• $\alpha = R_1 \sqsubseteq R_2, f = R_1(a, b)$ *and* $ga(R_2, a, b) \notin \mathcal{S}$*;*
• $\alpha = A_1 \sqsubseteq \neg A_2, f = A_1(a)$ *and* $\neg A_2(a) \notin \mathcal{S}$*;*
• $\alpha = \exists R \sqsubseteq \neg A, f = ga(R, a, b)$ *and* $\neg A(a) \notin \mathcal{S}$*;*
• $\alpha = R_1 \sqsubseteq \neg R_2, f = ga(R_1, a, b)$ *and* $\neg ga(R_2, a, b) \notin \mathcal{S}$*;*
• $\alpha = A \sqsubseteq \neg\exists R, f = A(a)$ *and there exists a constant* b *such that* $\neg ga(R, a, b) \notin \mathcal{S}$*;*
• $\alpha = \exists R_1 \sqsubseteq \neg\exists R_2, f = ga(R_1, a, b)$ *and there exists a constant* c *such that* $\neg ga(R_2, a, c) \notin \mathcal{S}$*;*
• $\alpha = (\text{funct } R), f = ga(R, a, b)$ *and there exists a constant* $x (x \neq b)$ *such that* $ga(R, a, x) \in \mathcal{S}$*,*
$=\!\!\!(b, x) \notin \mathcal{S}$*.*

Applicable inclusion or functionality can be used to construct *4-chase*$(\mathcal{K})$. Roughly speaking, *4-chase*$(\mathcal{K})$ is a (possibly infinite) set of membership asserions, constructed step-by-step starting from $\mathcal{A}$. At each step of construction, an inclusion assertion or a functionality assertion $\alpha \in \mathcal{T}$ is applied to a membership assertion $f \in \mathcal{S}$, which means adding a new suitable membership assertion to $\mathcal{S}$, thus obtaining a new set $\mathcal{S}'$ in which α is no longer applicable to f. For example, if $\alpha = A_1 \sqsubseteq \neg A_2$ is applicable to $f = A_1(a)$, then the membership assertion to be added to $\mathcal{S}$ is $\neg A_2(a)$, that is, $\mathcal{S}' = \mathcal{S} \cup \neg A_2(a)$.

In fact, Definition 3 extends Definition 4 of [5] in which only PIs in TBox are involved and Definition 8 of [12] in which only NIs and functionality assertions logically implied by TBox are involved, yet Definition 3 involves all axioms in TBox. Based on [5], we know that the construction process of *4-chase*$(\mathcal{K})$ strongly depends on the order in which we select the inclusion assertion or the functionality assertion to be applied at each step and the membership assertion to which such an inclusion assertion or a functionality assertion is applied, as well as on which constants we introduce at each step. Like Calvanese et al. have discussed in [5], we denote with Γ_A the set of all constant symbols occurring in $\mathcal{A}$ and we assume that we have an infinite set Γ_N of constant symbols not occurring in $\mathcal{A}$, such that the set $\Gamma_C = \Gamma_A \cup \Gamma_N$ is totally ordered in lexicographic way. And we select all axioms in $\mathcal{T}$, membership assertions, constants symbols in lexicographic order. Then our notion of *4-chase*$(\mathcal{K})$ is precisely given below.

Definition 4. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite ontology, let n be the number of membership assertions in $\mathcal{A}$, and let Γ_N be the set of constants defined above. Assume that the membership assertions in $\mathcal{A}$ are numbered from 1 to n following their lexicographic order, we have the definition as follows: $\mathcal{S}_0 = \mathcal{A}$ and $\mathcal{S}_{j+1} = \mathcal{S}_j \cup f_{new}$, where f_{new} is a membership assertion numbered with $n + j + 1$ in $\mathcal{S}_{j+1}$ and obtained as follows: Let f be the first membership assertion in $\mathcal{S}_j$ such that there exists an axiom $\alpha \in \mathcal{T}$ applicable in $\mathcal{S}_j$ to f. Let α be the lexicographically first inclusion or functionality assertion applicable in $\mathcal{S}_j$ to f. Let a_{new} be the constant of Γ_N that follows lexicographically all constants occurring in $\mathcal{S}_j$.*
Case *α, f of*
(cr1) $\alpha = A_1 \sqsubseteq A_2$ *and* $f = A_1(a)$ ***then*** $f_{new} = A_2(a)$
(cr2) $\alpha = A \sqsubseteq \exists R$ *and* $f = A(a)$ ***then*** $f_{new} = ga(R, a, a_{new})$

(cr3) $\alpha = \exists R \sqsubseteq A$ *and* $f = ga(R, a, b)$ ***then*** $f_{new} = A(a)$
(cr4) $\alpha = \exists R_1 \sqsubseteq \exists R_2$ *and* $f = ga(R_1, a, b)$ ***then*** $f_{new} = ga(R_2, a, a_{new})$
(cr5) $\alpha = R_1 \sqsubseteq R_2$ *and* $f = ga(R_1, a, b)$ ***then*** $f_{new} = ga(R_2, a, b)$
(cr6) $\alpha = A_1 \sqsubseteq \neg A_2$ *and* $f = A_1(a)$ ***then*** $f_{new} = \neg A_2(a)$
(cr7) $\alpha = \exists R \sqsubseteq \neg A$ *and* $f = ga(R, a, b)$ ***then*** $f_{new} = \neg A(a)$
(cr8) $\alpha = R_1 \sqsubseteq \neg R_2$ *and* $f = ga(R_1, a, b)$ ***then*** $f_{new} = \neg ga(R_2, a, b)$
(cr9) $\alpha = A \sqsubseteq \neg\exists R$ *and* $f = A(a)$ ***then*** $f_{new} = \neg ga(R, a, *)$
(cr10) $\alpha = \exists R_1 \sqsubseteq \neg\exists R_2$ *and* $f = ga(R_1, a, b)$ ***then*** $f_{new} = \neg ga(R_2, a, *)$
(cr11) $\alpha = (\text{funct } R)$ *and* $f = ga(R, a, b), \forall x, ga(R, a, x) \in \mathcal{S}_j$ *and* $b \neq x$ ***then*** $f_{new} = (\doteq(b, x))$.
Then 4-chase$(\mathcal{K})$ *is the set of membership assertions obtained as the infinite union of all* $\mathcal{S}_j$*, that is, 4-chase*$(\mathcal{K}) = \bigcup_{j \in N} \mathcal{S}_j$.

Note that not only PIs but also NIs and functionality assertions in $\mathcal{K}$ have a role in constructing *4-chase*$(\mathcal{K})$. Furthermore, each inclusion or functionality assertion in $\mathcal{T}$ can be applied at most once to a membership assertion (afterwards, the precondition is not satisfied and the inclusion or functionality assertion is no longer applicable). In the following, we will denote with *4-chase*$_i(\mathcal{K})$ the portion of the chase obtained after i applications of the chase rules, selected according to the ordering established in Definition 4, that is, *4-chase*$_i(\mathcal{K}) = \bigcup_{j \in \{0,\ldots,i\}} \mathcal{S}_j$. We can easily obtain the following proposition which is similar to Proposition 6 in [5].

Proposition 1. *Let* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ *be a DL-Lite ontology, and let* α *be an inclusion assertion or a functionality assertion in* $\mathcal{T}$*. Then, if there is an* $i \in N$ *such that* α *is applicable in 4-chase*$_i(\mathcal{K})$ *to a membership assertion* $f \in$ *4-chase*$_i(\mathcal{K})$*, then there is a* $j \geq i$ *such that 4-chase*$_{j+1}(\mathcal{K}) =$ *4-chase*$_j(\mathcal{K}) \cup f'$*, where* f' *is the result of applying* α *to* f *in 4-chase*$_j(\mathcal{K})$.

Proof. This proof is analogous to the proving process of Proposition 6 of [5]. Assume by contradiction that there is no $j > i$ such that *4-chase*$_{j+1}(\mathcal{K}) =$ *4-chase*$_j(\mathcal{K}) \cup f'$. Then we can deduce that either there are infinitely many membership assertions that precede f in the ordering that we choose for membership assertions in *4-chase*$(\mathcal{K})$, or that there are infinitely many chase rules applied to some membership assertions that precede f. However, none of these cases is possible. Indeed, f is assigned with an ordering number m such that exactly $m - 1$ membership assertions precede f. Furthermore, an inclusion assertion or a functionality assertion can be applied at most once to a membership assertion, and also there exists only a finite number l of inclusion assertions and functionality assertions. Therefore, it is possible to apply a chase rule to some membership assertion at most l times. We can thus conclude that the claim holds. □

Now, with the notion of *4-chase*$(\mathcal{K})$, we define the four-valued canonical interpretation denoted as *4-can*$(\mathcal{K}) = \langle \Delta^{\textit{4-can}(\mathcal{K})}, \cdot^{\textit{4-can}(\mathcal{K})} \rangle$, where

- $\Delta^{\textit{4-can}(\mathcal{K})} = \Gamma_C$,
- $a^{\textit{4-can}(\mathcal{K})} = a$, for each constant a occurring in *4-chase*$(\mathcal{K})$
- $A^{\textit{4-can}(\mathcal{K})} = \langle \{a \mid A(a) \in \textit{4-chase}(\mathcal{K})\}, \{b \mid \neg A(b) \in \textit{4-chase}(\mathcal{K})\} \rangle$, for each atomic concept A,
- $P^{\textit{4-can}(\mathcal{K})} = \langle \{(a_1, a_2) \mid P(a_1, a_2) \in \textit{4-chase}(\mathcal{K})\}, \{(b_1, b_2) \mid \neg P(b_1, b_2) \in \textit{4-chase}(\mathcal{K})\} \rangle$, for each atomic role P
- $\doteq^{\textit{4-can}(\mathcal{K})} = \langle \{(a_1, a_2) \mid \doteq(a_1, a_2) \in \textit{4-chase}(\mathcal{K})\} \cup \{(a, a) \mid \forall a \in \Gamma_C\}, \{(b_1, b_2) \mid \forall b_1, b_2 \in \Gamma_C$ and $b_1 \neq b_2\} \rangle$.

Like *can*$(\mathcal{K})$ in [5], we also define *4-can*$_i(\mathcal{K}) = \langle \Delta^{\textit{4-can}(\mathcal{K})}, .^{\textit{4-can}_i(\mathcal{K})}\rangle$, where $.^{\textit{4-can}_i(\mathcal{K})}$ is analogous to $.^{\textit{4-can}(\mathcal{K})}$ but it refers to *4-chase*$_i(\mathcal{K})$ instead of *4-chase*$(\mathcal{K})$. *4-chase*$(\mathcal{K})$ and *4-can*$(\mathcal{K})$ (resp., *4-chase*$_i(\mathcal{K})$ and *4-can*$_i(\mathcal{K})$) are strongly connected. It is easy to see that *4-can*$(\mathcal{K})$ (resp., *4-can*$_i(\mathcal{K})$) is also unique.

Now we give a four-valued interpretation *4-db*$(\mathcal{A}) = \langle \Delta^{\textit{4-db}(\mathcal{A})}, .^{\textit{4-db}(\mathcal{A})}\rangle$, where

- $\Delta^{\textit{4-db}(\mathcal{A})} = \Delta^{db(\mathcal{A})}$,
- $a^{\textit{4-db}(\mathcal{A})} = a$, for each constant a occurring in $\mathcal{A}$
- $A^{\textit{4-db}(\mathcal{A})} = \langle \{a \mid A(a) \in \mathcal{A}\}, \{\varnothing\}\rangle$, for each atomic concept A,
- $P^{\textit{4-db}(\mathcal{A})} = \langle \{(a_1, a_2) \mid P(a_1, a_2) \in \mathcal{A}\}, \{\varnothing\}\rangle$, for each atomic role P
- $=^{\textit{4-db}(\mathcal{A})} = \langle \{(a, a) \mid \forall a \in \Delta^{\textit{4-db}(\mathcal{A})}\}, \{(b_1, b_2) \mid \forall b_1, b_2 \in \Delta^{\textit{4-db}(\mathcal{A})} \text{ and } b_1 \neq b_2\}\rangle$.

Note that *4-can*$_0(\mathcal{K})$ is tightly related to the interpretation *4-db*$(\mathcal{A})$, that is, $.^{\textit{4-db}(\mathcal{A})} = .^{\textit{4-can}_0(\mathcal{K})}$.
The following theorem shows a notable property of *4-can*$(\mathcal{K})$.

Theorem 2. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ be a DL-Lite ontology. Then 4-can$(\mathcal{K})$ is a four-valued model of $\langle \mathcal{T}, \mathcal{A}\rangle$.*

Proof. First *4-can*$(\mathcal{K})$ satisfies all membership assertions in $\mathcal{A}$ because $\mathcal{A} \subseteq$ *4-chase*$(\mathcal{K})$. So we only need to prove that *4-can*$(\mathcal{K}) \models \mathcal{T}$. Let us proceed by contradiction considering all possible cases.

Suppose by contradiction that an inclusion assertion of the form $A_1 \sqsubseteq A_2 \in \mathcal{T}$, where A_1 and A_2 are atomic concepts, is not satisfied by *4-can*$(\mathcal{K})$. Then there exists a constant $a \in \Gamma_C$ such that $A_1(a) \in$ *4-chase*$(\mathcal{K})$ and $A_2(a) \notin$ *4-chase*$(\mathcal{K})$. However, such a situation would trigger the chase rule **cr1**, since $A_1 \sqsubseteq A_2$ would be applicable to $A_1(a)$ in *4-chase*$(\mathcal{K})$ and Proposition 1 ensures that such an inclusion assertion would be applied at some step in the construction of the chase, thus causing the insertion of $A_2(a)$ in *4-chase*$(\mathcal{K})$, hence contradicting the assumption. For the inclusion assertions such that $\exists R \sqsubseteq A$ can be proved in an analogous way.

Now suppose by contradiction that an inclusion assertion of the form $A \sqsubseteq \exists R \in \mathcal{T}$, where A and R are atomic concept and role respectively, is not satisfied by *4-can*$(\mathcal{K})$. Then there exists a constant $a \in \Gamma_C$ such that $A(a) \in$ *4-chase*$(\mathcal{K})$ and there does not exist a constant $c \in \Gamma_C$ such that $ga(R, a, c) \in$ *4-chase*$(\mathcal{K})$. However, such a situation would trigger the chase rule **cr2**, since $A \sqsubseteq \exists R$ would be applicable to $A(a)$ in *4-chase*$(\mathcal{K})$ and Proposition 1 ensures that such an inclusion assertion would be applied at some step in the construction of the chase, thus causing the insertion of $ga(R, a, c)$ in *4-chase*$(\mathcal{K})$, where $c \in \Gamma_C$ follows lexicographically all constants occurring in *4-chase*$(\mathcal{K})$ before the execution of **cr2**, hence contradicting the assumption. For the inclusion assertions such that $\exists R_1 \sqsubseteq \exists R_2$ can be proved in an analogous way.

Now we assume by contradiction that an inclusion assertion of the form $R_1 \sqsubseteq R_2 \in \mathcal{T}$, where R_1 and R_2 are atomic roles, is not satisfied by *4-can*$(\mathcal{K})$. Then there exists a pair of constants $a, b \in \Gamma_C$ such that $ga(R_1, a, b) \in$ *4-chase*$(\mathcal{K})$ and $ga(R_2, a, b) \notin$ *4-chase*$(\mathcal{K})$. However, such a situation would trigger the rule **cr5**, since $R_1 \sqsubseteq R_2$ would be applicable to $ga(R_1, a, b)$ in *4-chase*$(\mathcal{K})$ and Proposition 1 ensures that such an inclusion assertion would be applied at some step in the construction of the chase, thus causing the adding of $ga(R_2, a, b)$ in *4-chase*$(\mathcal{K})$ at some step, hence contradicting the assumption.

Next we assume by contradiction that an inclusion assertion of the form $A_1 \sqsubseteq \neg A_2 \in \mathcal{T}$, where A_1 and A_2 are atomic concepts, is not satisfied by *4-can*$(\mathcal{K})$. Then there exists a constant $a \in \Gamma_C$ such that $A_1(a) \in$ *4-chase*$(\mathcal{K})$ and $\neg A_2(a) \notin$ *4-chase*$(\mathcal{K})$. However, such a situation would trigger the rule **cr6**, since $A_1 \sqsubseteq \neg A_2$ would be applicable to $A_1(a)$ in *4-chase*$(\mathcal{K})$ and Proposition 1 ensures that such an inclusion assertion would be applied at some step in the construction of the chase, thus causing the adding of $\neg A_2(a)$ in *4-chase*$(\mathcal{K})$ at some step, hence contradicting the assumption. For the inclusion assertions such that $\exists R \sqsubseteq \neg A$ can be proved in an analogous way.

Now we assume by contradiction that an inclusion assertion of the form $\exists R_1 \sqsubseteq \neg\exists R_2 \in \mathcal{T}$, where R_1 and R_2 are atomic role, is not satisfied by *4-can*$(\mathcal{K})$. Then there exists a pair of constants $a, b \in \Gamma_C$ such that $ga(R_1, a, b) \in$ *4-chase*$(\mathcal{K})$ and there exist a constant $c \in \Gamma_C$ such that $\neg ga(R_2, a, c) \notin$ *4-chase*$(\mathcal{K})$. However, such a situation would trigger the rule **cr10**, since $\exists R_1 \sqsubseteq \neg\exists R_2$ would be applicable to $ga(R_1, a, b)$ in *4-chase*$(\mathcal{K})$ and Proposition 1 ensures that such an inclusion assertion would be applied at some step in the construction of the chase, thus causing the adding of $\neg ga(R_2, a, *)$ in *4-chase*$(\mathcal{K})$ at some step, hence contradicting the assumption. For the inclusion assertions such that $A \sqsubseteq \neg\exists R$ can be proved in an analogous way.

Now we assume by contradiction that an inclusion assertion of the form $R_1 \sqsubseteq \neg R_2 \in \mathcal{T}$, where R_1 and R_2 are atomic roles, is not satisfied by *4-can*$(\mathcal{K})$. Then there exists a pair of constants $a, b \in \Gamma_C$ such that $ga(R_1, a, b) \in$ *4-chase*$(\mathcal{K})$ and $\neg ga(R_2, a, b) \notin$ *4-chase*$(\mathcal{K})$. However, such a situation would trigger the rule **cr8**, since $R_1 \sqsubseteq \neg R_2$ would be applicable to $ga(R_1, a, b)$ in *4-chase*$(\mathcal{K})$ and Proposition 1 ensures that such an inclusion assertion would be applied at some step in the construction of the chase, thus causing the adding of $\neg ga(R_2, a, b)$ in *4-chase*$(\mathcal{K})$ at some step, hence contradicting the assumption.

Finally, assume by contradiction that a functionality assertion of the form (funct R), where R is a basic role, is not satisfied by *4-can*$(\mathcal{K})$. Then there exists constants $a, b, c \in \Gamma_C$ such that $ga(R, a, b)$, $ga(R, a, c) \in$ *4-chase*$(\mathcal{K})$ and $\neq(b, c) \notin$ *4-chase*$(\mathcal{K})$. However, such a situation would trigger the rule **cr11**, since (funct R) would be applicable to $ga(R, a, b)$ or $ga(R, a, c)$ in *4-chase*$(\mathcal{K})$ and Proposition 1 ensures that such a functionality assertion would be applied at some step in the construction of the chase, thus causing the adding of $\neq(b, c)$ in *4-chase*$(\mathcal{K})$ at some step, hence contradicting the assumption. $\square$

Theorem 2 tells us that for any *DL-Lite* ontology, we can always construct a four-valued model. In the following, we will use *4-can*$(\mathcal{K})$ for paraconsistent query answering over *DL-Lite* ontologies.

4.2 Query answering

In this section, we consider paraconsistent query answering over *DL-Lite* ontologies. We first give some properties which hold for *4-can*$(\mathcal{K})$. Then we show that paraconsistent query answering for UCQs over *DL-Lite* ontologies can be reduced to evaluate a finite reformulation of the query over *4-db*$(\mathcal{A})$.

Lemma 1. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite ontology, and let $\mathcal{M} = \langle \Delta^{\mathcal{M}}, \cdot^{\mathcal{M}} \rangle$ be a four-valued model for $\mathcal{K}$. Then, there is a function ψ from $\Delta^{\mathcal{M}}$ to $\Delta^{4\text{-}can(\mathcal{K})}$ such that*

(1) For each atomic concept A in $\mathcal{K}$ and each object $o \in \Delta^{4\text{-}can(\mathcal{K})}$, if $o \in proj^+(A^{4\text{-}can(\mathcal{K})})$, then $\psi(o) \in proj^+(A^{\mathcal{M}})$ and
(2) For each atomic role P in $\mathcal{K}$ and each pair of objects $o, o' \in \Delta^{4\text{-}can(\mathcal{K})}$, if $(o, o') \in proj^+(P^{4\text{-}can(\mathcal{K})})$, then $(\psi(o), \psi(o')) \in proj^+(P^{\mathcal{M}})$.

Proof. In order to define the function ψ, we start with the construction of *4-chase*$(\mathcal{K})$, and simultaneously show that properties (1) and (2) hold.

Base Step. For each constant d occurring in $\mathcal{A}$, we set $\psi(d^{4\text{-}can(\mathcal{K})}) = d^{\mathcal{M}}$ (notice that each model $\mathcal{M}$ interprets each such constant with an element in $\Delta^{\mathcal{M}}$). We remember that $chase_0(\mathcal{K}) = \mathcal{A}$, $\Delta^{can_0(\mathcal{K})} = \Delta^{4\text{-}can(\mathcal{K})} = \Gamma_C$, and that, for each constant d occurring in $\mathcal{A}$, $d^{can_0(\mathcal{K})} = d$. Then, it is easy to see that for each object $o_c \in \Delta^{can_0(\mathcal{K})}$ (resp., for each pair of objects $o_c^1, o_c^2 \in \Delta^{can_0(\mathcal{K})}$), if $o_c \in proj^+(A^{can_0(\mathcal{K})})$, where A is an atomic concept in $\mathcal{K}$ (resp., $o_c^1, o_c^2 \in proj^+(P^{can_0(\mathcal{K})})$, where P is an atomic role in $\mathcal{K}$), we have that $A(o_c) \in chase_0(\mathcal{K})$ (resp., $P(o_c^1, o_c^2) \in chase_0(\mathcal{K})$). Since $\mathcal{M}$ satisfies all membership assertions in $\mathcal{A}$, we also have that $\psi(o_c) \in proj^+(A^{\mathcal{M}})$ (resp., $(\psi(o_c^1), \psi(o_c^2)) \in proj^+(P^{\mathcal{M}})$).

Inductive Step. Based on the construction of *4-chase*$(\mathcal{K})$ and *4-can*$(\mathcal{K})$, we know that for each atomic concept A (resp., each atomic role P), $proj^+(A^{\textit{4-can}(\mathcal{K})})$ (resp., $proj^+(R^{\textit{4-can}(\mathcal{K})})$) only relate with PMAs in *4-chase*$(\mathcal{K})$. And that, PMAs in *4-chase*$(\mathcal{K})$ only includes $\mathcal{A}$ and other PMAs obtained by applying rule **cr1**-**cr5**, whereas PMAs can not be obtained by applying other rule **cr6**-**cr11**. Now let us assume that $chase_{i+1}(\mathcal{K})$ is obtained from $chase_i(\mathcal{K})$ by applying rule **cr2**. This means that an inclusion assertion of the form $A \sqsubseteq \exists R$, where A is an atomic concept in $\mathcal{T}$, and R is a basic role in $\mathcal{T}$, is applied in $chase_i(\mathcal{K})$ to a membership assertion of the form $A(d)$, such that there does not exist a constant $f \in \Gamma_C$ such that $ga(R, d, f) \in chase_i(\mathcal{K})$. Therefore $chase_{i+1}(\mathcal{K}) = chase_i(\mathcal{K}) \cup \{ga(R, d, e)\}$, where e follows lexicographically all constants appearing in $chase_i(\mathcal{K})$. By the induction hypothesis, there exists $o_m \in \Delta^{\mathcal{M}}$ such that $\psi(d) = o_m$ and $o_m \in proj^+(A^{\mathcal{M}})$. Because $\mathcal{M}$ is a four-valued model of $\mathcal{K}$, so $\mathcal{M}$ satisfies $A \sqsubseteq \exists R$, that is, there is at least one object $o^{'}_m \in \Delta^{\mathcal{M}}$ such that $(o_m, o^{'}_m) \in proj^+(R^{\mathcal{M}})$. Then, we set $\psi(e) = o^{'}_m$, and we can conclude that $(\psi(d), \psi(e)) \in proj^+(R^{\mathcal{M}})$. With an analogous argument we can prove the inductive step also in those cases in which $chase_{i+1}(\mathcal{K})$ is obtained from $chase_i(\mathcal{K})$ by applying one of the rules **cr1**, **cr3**, **cr4** or **cr5**. □

From Lemma 1 we know that, for every four-valued model $\mathcal{M}$ of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, there is a homomorphism from *4-can*$(\mathcal{K})$ to $\mathcal{M}$ that maps the objects which support concepts and roles to be true in *4-can*$(\mathcal{K})$ to objects which support concepts and roles to be true in $\mathcal{M}$. We know that for a union of conjunctive queries q over an onotology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, a *certain answer* to q in $\mathcal{K}$ is only related with the constants which support concepts and roles to be true in every four-valued model of $\mathcal{K}$. So we have the following theorem.

Theorem 3. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite ontology, and let Q be a union of conjunctive queries over $\mathcal{K}$. Then 4-Ans$(q, \mathcal{K}) = Q^{\textit{4-can}(\mathcal{K})}$, where $Q^{\textit{4-can}(\mathcal{K})}$ is similar as q^I defined in Definition 2.*

Proof. First we know that $\Delta^{\textit{4-can}(\mathcal{K})} = \Gamma^C$ and that, for each constant d occurring in $\mathcal{K}$, $d^{\textit{4-can}(\mathcal{K})} = d$. Therefore, given a tuple $\vec{t}$ of constants occurring in $\mathcal{A}$, we have that $\vec{t}^{\textit{4-can}(\mathcal{K})} = \vec{t}$. So we can hence rephrase the claim as $\vec{t} \in$ *4-Ans*$(q, \mathcal{K})$ iff $\vec{t} \in Q^{\textit{4-can}(\mathcal{K})}$.

"$\Rightarrow$" Suppose $\vec{t} \in$ *4-Ans*$(q, \mathcal{K})$. Then, since *4-can*$(\mathcal{K})$ is a model of $\mathcal{K}$, we have that $\vec{t} \in Q^{\textit{4-can}(\mathcal{K})}$.

"$\Leftarrow$" Suppose $\vec{t} \in Q^{\textit{4-can}(\mathcal{K})}$. Let Q be the union of conjunctive queries $Q = \{q_1, \cdots, q_k\}$ with q_i defined as $q_i(\vec{x}) \leftarrow conj_i(\vec{x}, \vec{y_i})$ for each $i \in \{1, \cdots, k\}$. Then there exists $i \in \{1, \cdots, k\}$ such that there is a homomorphism from $conj_i(\vec{t}, \vec{y_i})$ to *4-can*$(\mathcal{K})$; that is, there exists an assignment $\mu : V \rightarrow \Delta^{\textit{4-can}(\mathcal{K})}$ that maps the variables V occurring in $conj_i(\vec{t}, \vec{y_i})$ to objects of $\Delta^{\textit{4-can}(\mathcal{K})}$, such that all atoms in $conj_i(\vec{t}, \vec{y_i})$ under the assignment μ evaluate to t or $\mathbb{B}$ in *4-can*$(\mathcal{K})$.

Now let $\mathcal{M}$ be a model for $\mathcal{K}$. By Lemma 1, there is a homomorphism ψ from $\Delta^{\textit{4-can}(\mathcal{K})}$ to $\Delta^{\mathcal{M}}$. Consequently, the function obtained by composing ψ and μ is a function that maps the variables V occurring in $conj_i(\vec{t}, \vec{y_i})$ to objects of $\Delta^{\mathcal{M}}$, such that all atoms in $conj_i(\vec{t}, \vec{y_i})$ under the assignment μ evaluate to t or $\mathbb{B}$ in $\mathcal{M}$. Therefore, $\vec{t}^{\mathcal{M}} \in Q^{\mathcal{M}}$, which implies that $\vec{t} \in$ *4-Ans*$(Q, \mathcal{K})$. □

Theorem 3 tells us that for any union of conjunctive queries Q, the answers to Q over $\mathcal{K}$ correspond to the evaluation of Q in *4-can*$(\mathcal{K})$.

Theorem 4. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite ontology, and let Q be a union of conjunctive queries over $\mathcal{K}$. Then 4-Ans$(Q, \mathcal{K}) = \bigcup_{q_i \in Q}$ 4-Ans$(q_i, \mathcal{K})$.*

Proof. Suppose $\vec{t} \in$ *4-Ans*$(Q, \mathcal{K})$, and suppose that every q_i is of the form $q_i(\vec{x}) \leftarrow conj_i(\vec{x}, \vec{y_i})$ for each $q_i \in Q$. Then by Theorem 3, $\vec{t}^{\textit{4-can}(\mathcal{K})} \in Q^{\textit{4-can}(\mathcal{K})}$, which implies that there exists $i \in \{1, \cdots, k\}$ such that $\vec{t}^{\textit{4-can}(\mathcal{K})} \in conj_i(\vec{t}, \vec{y_i})^{\textit{4-can}(\mathcal{K})}$. Hence, from Theorem 3, it follows that $\vec{t} \in$ *4-Ans*$(q_i, \mathcal{K})$. □

Theorem 4 states that the set of answers to a union of conjunctive queries Q in $\mathcal{K}$ corresponds to the union of the answers to the various conjunctive queries in Q. We now extend the results in [5] by considering generic *DL-Lite* ontologies (either satisfiable or unsatisfiable), and provide the following theorem which shows that query answering for UCQs over *DL-Lite* ontologies can be reduced to evaluate a finite reformulation of the query over *4-db*$(\mathcal{A})$.

Theorem 5. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be an inconsistent DL-Lite ontology, q a conjunctive query over $\mathcal{K}$, and let PR be a union of conjunctive queries returned by* **PerfectRef**$(q, \mathcal{T})$. *Then 4-Ans$(q, \mathcal{K}) = PR^{\textit{4-db}(\mathcal{A})}$, where $PR^{\textit{4-db}(\mathcal{A})}$ is similar as q^I defined in Definition 2.*

Proof. The proof is a consequence of Lemma 39 in [5] by extending the classical semantics to four-valued semantics. □

Based on Theorem 5, we have **Algorithm Answer**$(Q, \mathcal{K})$ as follows:
Input: a *DL-Lite* ontology $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, an UCQ Q;
Output: *4-Ans*$(Q, \mathcal{K})$;
Return: $(\bigcup_{q_i \in Q} \text{PerfectRef}(q_i, \mathcal{T}))^{\textit{4-db}(\mathcal{A})}$.
In fact, **Algorithm Answer**$(Q, \mathcal{K})$ first computes the perfect reformulation PR of Q by rewriting rule PerfectRef, then returns PR$^{\textit{4-db}(\mathcal{A})}$ directly. The following theorem shows the correctness of the algorithm.

Theorem 6. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite ontology, and Q be a union of conjunctive queries, and $\vec{t}$ a tuple of constants in $\mathcal{K}$. We have: (1) the algorithm $Answer(Q, \mathcal{K})$ terminates; (2) $\vec{t} \in$ 4-Ans$(Q, \mathcal{K})$ if and only if $\vec{t} \in Answer(Q, \mathcal{K})$*

Proof. It is immediately to obtain (1) because $Answer(Q, \mathcal{K})$ mainly depends on the algorithm PerfectRef$(q, \mathcal{T})$ which is terminates [5]. Through Theorem 4 and Theorem 5 we can easily obtain (2). □

Example 4 (Example 2 contd.). *Let us consider a query $q(x) = Stud(x) \wedge \exists y.hasTutor(x, y)$. By executing $Answer(q, \mathcal{K})$, it first execute algorithm PerfectRef$(q, \mathcal{T})$ which return the union of the following conjunctive queries: $\{Stud(x) \wedge \exists y.hasTutor(x, y)$, $PhDStud(x) \wedge \exists y.hasTutor(x, y)$, $PhDStud(x) \wedge Stud(x)$, $PhDStud(x)\}$,then based on the interpretation 4-db$(\mathcal{A})$, we obtain that 4-Ans$(q, \mathcal{K})$ is $\{a\}$.*

The following theorem shows the complexity of paraconsistent query answering over *DL-Lite* ontologies:

Theorem 7. *Paraconsistent answering unions of conjunctive queries in DL-Lite $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is PTime in the size of the TBox, and* LOGSPACE *in the size of the ABox.*

Proof. First the algorithm PerfectRef$(q, \mathcal{T})$ runs in time polynomial in the size of $\mathcal{T}$ [5]. Because unions of conjunctive queries are a subclass of FOL queries [5], and also the evaluation of a union of conjunctive queries over a database can be computed in LogSpace with respect to the size of the database. So the claim holds. □

5 Conclusion and Future work

Paraconsistent query answering over ontologies is an important problem in ontology engineering. In this paper, we considered the problem of answering unions of conjunctive queries posed to inconsistent *DL-Lite* ontologies. We first gave a four-valued semantics for *DL-Lite*. Then through introducing the definition of *4-chase*$(\mathcal{K})$, we gave a four-valued canonical interpretation *4-can*$(\mathcal{K})$ for any inconsistent

DL-Lite ontology $\mathcal{K}$ and we proved this interpretation was also a four-valued model of $\mathcal{K}$. Then we presented a theorem, that is, for any union of conjunctive queries Q, the answers to Q over an inconsistent *DL-Lite* ontology $\mathcal{K}$ correspond to the evaluation of Q over *4-can*$(\mathcal{K})$. Furthermore, based on the properties of *4-can*$(\mathcal{K})$ and *4-db*$(\mathcal{A})$, we proved that query answering for UCQs over inconsistent *DL-Lite* ontologies can be reduced to evaluate a finite reformulation of the query over *4-db*$(\mathcal{A})$ which is also our motivation to propose a tractable algorithm to compute the certain answers to a query over an inconsistent *DL-Lite* ontology. Our algorithm first computes the perfect reformulation PR of Q by rewriting rule PerfectRef [5], then returns $\text{PR}^{\textit{4-db}(\mathcal{A})}$ directly. As a future work, we will implement our algorithm and provide experimental results and we also plan to do the work of paraconsistent query answering over $\textit{DL-Lite}_{\mathcal{R},\sqcap}$ and $\textit{DL-Lite}_{\mathcal{F},\sqcap}$.

Acknowledgments

Our research is supported by the National Grand Fundamental Research 973 Program of China under grant (No.2007CB307100, No.2007CB307106) and by the Specialized Research Foundation of Doctoral Program of Higher Education of China under grant(No.20050004008).

References

[1] M. Arenas, L. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of the 18th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 68–79. ACM, 1999.

[2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. Description logic terminology. In F. B. et al., editor, *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[4] A. Calì, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proceedings of the 22th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 260–271. ACM, 2003.

[5] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[6] Z. Huang, F. van Harmelen, and A. ten Teije. Reasoning with inconsistent ontologies. In L. P. Kaelbling and A. Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 454–459, 2005.

[7] D. Lembo and M. Ruzzi. Consistent query answering over description logic ontologies. In M. Marchiori, J. Pan, and C. de Sainte Marie, editors, *Proceedings of the 1st International Conference on Web Reasoning and Rule Systems*, pages 194–208, 2007.

[8] Y. Ma, P. Hitzler, and Z. Lin. Algorithms for paraconsistent reasoning with OWL. In E. Franconi, M. Kifer, and W. May, editors, *Proceedings of the 4th European Semantic Web Conference*, pages 399–413, 2007.

[9] Y. Ma, P. Hitzler, and Z. Lin. Paraconsistent reasoning for expressive and tractable description logics. In F. Baader, C. Lutz, and B. Motik, editors, *Description Logics*, 2008.

[10] P. F. Patel-Schneider. A four-valued semantics for terminological logics. *Artif. Intell.*, (3):319–351, 1989.

[11] J. Villadsen. Paraconsistent query answering systems. In T. Andreasen, A. Motro, H. Christiansen, and H. L. Larsen, editors, *Proceedings of the 5th International conference on Flexible Query Answering Systems*, pages 370–384, 2002.

[12] L. Zhou, H. Huang, G. Qi, Y. Ma, Z. Huang, and Y. Qu. Measuring inconsistency degrees of *DL-Lite* ontologies. In *Proceedings of 2009 IEEE/WIC/ACM International Conference on Web Intelligence (WI-09)*, Accepted, 2009.