Ann Math Artif Intell (2017) 79:45–75
DOI 10.1007/s10472-015-9494-2

Prioritized assertional-based removed sets revision of *DL-Lite* belief bases

Salem Benferhat[1] · Zied Bouraoui[1] · Odile Papini[2] · Eric Würbel[2]

Published online: 28 December 2015

Abstract In real world applications, information is often provided by multiple sources having different priority levels reflecting for instance their reliability. This paper investigates ”Prioritized Removed Sets Revision” (PRSR) for revising stratified *DL-Lite* knowledge bases when a new sure piece of information, called the input, is added. The strategy of revision is based on inconsistency minimization and consists in determining smallest subsets of assertions (prioritized removed sets) that should be dropped from the current stratified knowledge base in order to restore consistency and accept the input. We consider different forms of input: A membership assertion, a positive or a negative inclusion axiom. To characterize our revision approach, we first rephrase Hansson's postulates for belief bases revision within a *DL-Lite* setting, we then give logical properties of PRSR operators. In some situations, the revision process leads to several possible revised knowledge bases where defining a selection function is required to keep results within *DL-Lite* fragment. The last part of the paper shows how to use the notion of hitting set in order to compute the PRSR outcome. We also study the complexity of PRSR operators, and show that, in some cases, the computational complexity of the result can be performed in polynomial time.

This paper is a revised and extended version of the conference papers [7–9].

✉ Zied Bouraoui
bouraoui@cril.fr

Salem Benferhat
benferhat@cril.univ-artois.fr

Odile Papini
odile.papini.@univ-amu.fr

Eric Würbel
eric.wurbe.l@univ-amu.fr

[1] Artois University - Nord de France, CRIL CNRS UMR 8188 Artois, 62307 Lens, France

[2] Aix-Marseille University, LSIS-CNRS UMR 7296, 13288 Marseille, France

Keywords Belief revision · Prioritized knowledge bases · Lightweight ontologies

Mathematics Subject Classification (2010) 68T27 · 68T30

1 Introduction

In the past few years, there has been an increasing use of ontologies in many application areas. Description Logics (DLs) have been recognized as a powerful formalism for representing and reasoning with ontologies [3]. A DL knowledge base is built upon two distinct components: A terminological base (called TBox), representing generic knowledge about the application domain, and an assertional base (called ABox), containing extentional knowledge (i.e. facts, individuals or constants) that instantiate terminological knowledge. Recently, attention has been paid to *DL-Lite*, a family of tractable DLs investigated in [20]. *DL-Lite* provides a powerful framework that allows for a flexible representation of knowledge with a low computational complexity for the reasoning process [2]. In particular, *DL-Lite* is specifically tailored for applications that use a huge volume of data, like Web applications, in which query answering is the most important reasoning task.

For this reason, *DL-Lite* is well suited for various application areas such as the Semantic Web, where *DL-Lite* provides the logical underpinning of the *OWL2-QL* [1] language. In particular, *DL-Lite* is fitted towards Ontology-Based Data Access (OBDA) in which the TBox is used to reformulate asked queries in order to offer a better access to the set of assertions stored in the ABox [50].

Originally, DLs have been introduced to represent the static aspects of a domain of interest [5]. However, for some applications, knowledge may not be static and evolves from a situation to another in order to cope with changes that occur over time. Such dynamic aspects have been recognized as an important problem (e.g. [22, 43, 53, 61]) and often concern the situation where new information should be taken into account in order to modify an old one while ensuring the consistency of the result. Such problem is well-known as a belief revision problem. It has been defined as a knowledge change operation and was characterized for instance by the well-known AGM postulates [1]. These postulates focus on the logical structure of knowledge and are based on three main ideas: *(i)* The principle of priority which states that the priority between beliefs is given to new pieces of information, *(ii)* the consistency principle which states that the result of the revision operation must be a consistent set of beliefs, and *(iii)* the principle of minimal change which states that the least possible initial beliefs should be changed during the revision operation. Note that AGM postulates were defined in the context of the revision of belief sets, i.e. deductively closed sets of formulas, possibly infinite. Besides, an axiomatic characterization for revising belief bases, i.e. finite sets of formulas was given in [28, 37].

Recently, several works have been proposed for revising DLs knowledge bases. In [25, 26] an adaptation of the AGM theory was discussed in order to accommodate it to DLs. In [33, 52, 60] an extension of kernel-based revision and semi-revision operators to DLs frameworks has been proposed which is closely related to the one proposed by [36] in a propositional logic setting. In [59] the concept of "debugging" terminological bases has been introduced. The proposed solutions mainly adapt what has been proposed in diagnosis

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

to general terminological knowledge bases. Regarding *DL-Lite* knowledge bases, few research works have been proposed for the revision problem. In [51, 58, 61], model-based approaches for revising DLs have been proposed. In [22, 42] a computational complexity analysis has been given for revising DL knowledge bases. In [22], a formula-based approach for revising *DL-Lite* knowledge bases has been presented. Two algorithms have been provided: One for revising the TBox, and the other for revising the ABox. Another operator for ABox revision in *DL-Lite* based on graph structure has been introduced in [30] where new information is restricted to ABox assertions.

Belief revision has been largely considered in the literature when knowledge bases are encoded using a propositional language. Among these revision approaches the so-called Removed Sets Revision, also known as a cardinality-based approach, has been proposed in [12, 49] for revising a set of propositional formulas. This approach stems from removing a minimal number of formulas, called removed set, to restore consistency. The minimality in Removed Sets Revision refers to the cardinality criterion and not to the set-inclusion one. This approach has interesting properties: It has not a high computational complexity, it is not too cautious and satisfies all rational AGM postulates when extended to belief sets revision.

Besides, data are often provided by several and potentially conflicting sources. Concatenating them may lead to a prioritized or a stratified ABox. This stratification generally results from two situations as pointed out e.g. in [13, 14]. The first one is when each source provides its set of data without any priority between them, but there exists a total pre-ordering between different sources reflecting their reliability. The other one is when the sources are considered as equally reliable (i.e. having the same reliability level), but there exists a preference ranking between the set of provided data according to their level of certainty. The role of priorities in belief revision is very important and was largely studied in the literature where knowledge bases are encoded in a propositional logic setting (e.g. [15, 16]). The notion of priorities in DLs is used in (e.g. [4, 54, 56]) to deal with defaults terminology while assuming that the ABox is completely certain. Moreover, it is also used in the context of inconsistency handling, e.g. [11, 18, 40, 44]]. However, as far as we know, very few works address the revision of prioritized DLs knowledge bases (e.g. [10, 55]).

This paper studies Prioritized Removed Sets Revision (PRSR), when knowledge bases are described in *DL-Lite* logics. One of the motivations in considering PRSR is to take advantage of the tractability of *DL-Lite* for the revision process as well as of rational properties satisfied by PRSR. In particular, we investigate the well-known $DL\text{-}Lite_{R}$ logic which offers a good compromise between expressive power and computational complexity. We consider different forms of input: A membership assertion, a positive inclusion axiom or a negative inclusion axiom, since they lead to different revision problems, different algorithms and different complexity results. A crucially important problem that arises when revising a *DL-Lite* knowledge base is how to restore consistency. In this paper restoring consistency leads to ignoring some assertions, namely the priority is given to the TBox over the ABox. Another important feature when dealing with *DL-Lite* knowledge bases is that computing the set of minimal information responsible of inconsistency can be done in polynomial time. Besides minimal assertional sets that cause inconsistency are either singletons or doubletons. This is helpful for the definition of removed sets necessary to restore consistency in presence of new information.

The rest of this paper is organized as follows. Section 2 gives brief preliminaries on *DL-Lite* logics. Section 3 studies Prioritized Removed Sets Revision within this framework when priorities between assertional facts are available. Section 4 reformulates the well-known Hansson's postulates defined for propositional belief bases revision within a *DL-Lite* setting and gives logical properties of PRSR operators. Section 5 provides algorithms for computing prioritized removed sets through the use of hitting sets. Section 6 presents a study on the computational complexity of PRSR operators. Finally, Section 7 presents some related works and Section 8 concludes the paper.

2 A refresher on DL-Lite logic

2.1 Syntax

The language of $DL\text{-}Lite_{core}$ is the core language for $DL\text{-}Lite_R$ [21]. It is defined as follows:

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

where A is an atomic concept, P is an atomic role and P^- is the inverse of an atomic role. Concepts B (*resp.* C) are called basic (*resp.* complex) concepts and roles R (r*esp.* E) are called basic (*resp.* complex) roles.

A *DL-Lite* knowledge base is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is called the TBox (Terminological Box) and $\mathcal{A}$ is called the ABox (Assertional Box). A $DL\text{-}Lite_{core}$ TBox consists of a finite set of *inclusion axioms* of the form:

$$B \sqsubseteq C.$$

The $DL\text{-}Lite_{core}$ ABox consists of a finite set of *membership assertions* on atomic concepts and on atomic roles respectively of the form:

$$A(a) and P(a, b),$$

where a and b are two individuals. $DL\text{-}Lite_R$ extends $DL\text{-}Lite_{core}$ with the ability to specify in the TBox inclusion axioms between roles of the form:

$$R \sqsubseteq E.$$

For the sake of simplicity, we only consider $DL\text{-}Lite_R$ that underlies *OWL2-QL*. However results of this work can be easily adapted to other *DL-Lite* variants. Among them, $DL\text{-}Lite_F$ might be of interest. It extends $DL\text{-}Lite_{core}$ with the ability to specify functionnality property on roles (or their inverse). For more details about the other members of the *DL-Lite* family see [2].

In the rest of this paper, when there is no ambiguity, we simply use *DL-Lite* instead of $DL\text{-}Lite_R$.

2.2 Semantics

The semantics of *DL-Lite* is given in terms of interpretations. An interpretation $\mathcal{I}$=($\Delta^{\mathcal{I}}$, $.^{\mathcal{I}}$) consists of a non-empty domain $\Delta^{\mathcal{I}}$ and an interpretation function $.^{\mathcal{I}}$. The function $.^{\mathcal{I}}$ assigns to each individual a an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, to each concept C a subset $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to each role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function $.^{\mathcal{I}}$ is extended to all constructs of *DL-Lite* as follows:

$$\begin{aligned} A^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}}, \\ (P)^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}, \\ (P^{-})^{\mathcal{I}} &= \{(y,x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} | (x,y) \in P^{\mathcal{I}}\}, \\ (\exists R)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} | \exists y \in \Delta^{\mathcal{I}} \text{ such that } (x,y) \in R^{\mathcal{I}}\}, \\ (\neg B)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus B^{\mathcal{I}}, \\ (\neg R)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}. \end{aligned}$$

For the TBox, we say that $\mathcal{I}$ satisfies a concept (*resp.* role) inclusion axiom, denoted by $\mathcal{I} \models B \sqsubseteq C$ (*resp.* $\mathcal{I} \models R \sqsubseteq E$), if and only if $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ (*resp.* $R^{\mathcal{I}} \subseteq E^{\mathcal{I}}$). For the ABox, we say that $\mathcal{I}$ satisfies a concept (*resp.* role) membership assertion, denoted by $\mathcal{I} \models A(a)$ (*resp.* $\mathcal{I} \models P(a,b)$), if and only if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (*resp.* $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$). Note that we only consider *DL-Lite* with unique name assumption. An interpretation $\mathcal{I}$ is said to satisfy a knowledge base $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ if and only if $\mathcal{I}$ satisfies every axiom in $\mathcal{T}$ and every assertion in $\mathcal{A}$. Such an interpretation called a model of $\mathcal{K}$ [21].

2.3 Incoherence, inconsistency and negative closure

Two kinds of inconsistency can be distinguished in DL-based knowledge bases: *Incoherence* and *inconsistency* [5, 24]. The former is considered as a kind of inconsistency in the TBox, i.e. the terminological part of a knowledge base. The latter is the classical inconsistency for knowledge bases. Namely, a knowledge base is said to be inconsistent if and only if it does not admit any model and it is said to be incoherent if there exists at least a non-satisfiable concept (i.e. no individual can belong to the concept).

Definition 1 A *DL-Lite* terminological base $\mathcal{T}$ is said to be incoherent if there exists a concept C (resp. a role R) such that for each interpretation $\mathcal{I}$ which is a model of $\mathcal{T}$, we have $C^{\mathcal{I}} = \emptyset$ (resp. $R^{I} = \emptyset$).

A minimal example of incoherent TBox is the one composed of the two inclusion axioms $\mathcal{T} = \{B_1 \sqsubseteq B_2, B_1 \sqsubseteq \neg B_2\}$. One can easily check that for all models $\mathcal{I}$ of $\mathcal{T}$ we have $B_1^{\mathcal{I}} = \emptyset$. In a propositional setting, the counterpart of incoherence is a so-called potential inconsistency, as defined for instance in [48]. The concept of knowledge base inconsistency is defined by:

Definition 2 A *DL-Lite* knowledge base $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ is said to be inconsistent if it does not admit any model.

In *DL-Lite*, a TBox $\mathcal{T}$={PIs,NIs} can be viewed as composed of positive inclusion axioms, denoted by (PIs), and negative inclusion axioms, denoted by (NIs). PIs are of the form $B_1 \sqsubseteq B_2$ or $R_1 \sqsubseteq R_2$ and NIs are of the form $B_1 \sqsubseteq \neg B_2$ or $R_1 \sqsubseteq \neg R_2$.

The negative closure of $\mathcal{T}$, denoted by $cln(\mathcal{T})$, performs interaction between PIs and NIs. It represents the propagation of the NIs using both PIs and NIs in the TBox. $cln(\mathcal{T})$ is obtained using the following rules repeatedly until reaching a fixed point (see [21] for more details):

– all *NIs* in $\mathcal{T}$ are in $cln(\mathcal{T})$;

- if $B_1 \sqsubseteq B_2$ is in $\mathcal{T}$ and $B_2 \sqsubseteq \neg B_3$ or $B_3 \sqsubseteq \neg B_2$ is in $cln(\mathcal{T})$, then $B_1 \sqsubseteq \neg B_3$ is in $cln(\mathcal{T})$;
- if $R_1 \sqsubseteq R_2$ is in $\mathcal{T}$ and $\exists R_2 \sqsubseteq \neg B$ or $B \sqsubseteq \neg \exists R_2$ is in $cln(\mathcal{T})$, then $\exists R_1 \sqsubseteq \neg B$ is in $cln(\mathcal{T})$;
- if $R_1 \sqsubseteq R_2$ is in $\mathcal{T}$ and $\exists R_2^- \sqsubseteq \neg B$ or $B \sqsubseteq \neg \exists R_2^-$ is in $cln(\mathcal{T})$, then $\exists R_1^- \sqsubseteq \neg B$ is in $cln(\mathcal{T})$;
- if $R_1 \sqsubseteq R_2$ is in $\mathcal{T}$ and $R_2 \sqsubseteq \neg R_3$ or $R_3 \sqsubseteq \neg R_2$ is in $cln(\mathcal{T})$, then $R_1 \sqsubseteq \neg R_3$ is in $cln(\mathcal{T})$;
- if one of the assertions $\exists R \sqsubseteq \neg \exists R$, $\exists R^- \sqsubseteq \neg \exists R^-$ or $R \sqsubseteq \neg R$ is in $cln(\mathcal{T})$ then all these assertions are in $cln(\mathcal{T})$.

An important property has been established in [21] for consistency checking in *DL-Lite*. Formally, $\mathcal{K}$ is consistent if and only if $\langle cln(\mathcal{T}), \mathcal{A}\rangle$ is consistent [21].

2.4 Conjunctive queries

An n-ary query is an open formula of First-Order Logic (FOL) with equalities of the form

$$q = \{\mathbf{x} \,|\phi(\mathbf{x})\},$$

where $\phi(\mathbf{x})$ is a FOL formula with free variables $\mathbf{x}$=$(x_1, ..., x_n)$ (called also answer variables) and the arity n of q is the number of its free variables. When n=0, the query is said to be a boolean or ground query.

Given an interpretation $\mathcal{I}$=$(\Delta^{\mathcal{I}}, .^{\mathcal{I}})$, a boolean query is either interpreted as true in $\mathcal{I}$ if $[\phi]^I = true$ or false if $[\phi]^I = false$. Indeed, the answer to such a query is either "yes" or "no". When $n > 0$, A non-boolean query q is interpreted as the set of tuples of the domain elements, called answers sets with respect to $\mathcal{I}$, such that, if we substitute $\mathbf{x}$ by an answer set $\mathbf{a}$ the query q will be evaluated to true in $\mathcal{I}$. Namely $q^{\mathcal{I}} = \{\mathbf{a}_i \in (\Delta^{\mathcal{I}})^n | [\phi(\mathbf{a}_i)]^I = true\}$. An interpretation that evaluates a boolean query (*resp.* non-boolean query) to true (*resp.* to a non empty answers set), is said to be a model of that query, written $\mathcal{I} \models q$.

Within *DL-Lite*, the most interesting queries are the class of conjunctive queries and the class of union of conjunctive queries. A Conjunctive Query (CQ) is a query of the form:

$$q = \{\mathbf{x} \mid \exists \mathbf{y}.conj(\mathbf{x}, \mathbf{y})\},$$

where $\mathbf{x}$ are free variables called distinguished or answer variables, $\mathbf{y}$ are existentially quantified variables called non-distinguished or bounded variables, and $conj(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms of the form $A(t_i)$ or $P(t_i, t_j)$ and equalities, where the predicates A and P are respectively an atomic concept and an atomic role name appearing in $\mathcal{K}$, and t_i, t_j are terms, i.e constants (individuals) in $\mathcal{A}$ or variables occuring in $\mathbf{x}$ or $\mathbf{y}$. Notice that we call instance query a query consisting of a single atom with no free variable, namely an ABox assertion. A Union of Conjunctive Query (UCQ) denoted by Q is simply is an expression of the form:

$$Q = \{\mathbf{x} \mid \bigvee_{i=i,..,n} \exists \mathbf{y}_i.conj(\mathbf{x}, \mathbf{y}_i)\}.$$

where each $conj(\mathbf{x}, \mathbf{y}_i)$ is a conjunction of atoms and equalities with answer variables $\mathbf{x}$ and bound variables $\mathbf{y}_i$. Obviously, the class of UCQ contains the one of CQ.

Given $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ a *DL-Lite* knowledge base and a CQ q, we write $\mathcal{K} \models q$ when $\mathcal{I} \models q$ for all models $\mathcal{I}$ of $\mathcal{K}$, otherwise $\mathcal{K} \not\models q$. The answer to q over $\mathcal{K}$, denoted $ans(q, \mathcal{K})$, is the set of tuples of constants appearing in $\mathcal{K}$ such that $\forall \mathbf{a_i}$: $\mathbf{a_i}^{\mathcal{I}} \in q^{\mathcal{I}}$, for every model $\mathcal{I}$ of $\mathcal{K}$. Namely $ans(q, \mathcal{K}) = \{\mathbf{a_i} \in (\mathcal{K})^n | \mathcal{K} \models q(\mathbf{a_i})\}$ where $q(\mathbf{a_i})$ is the closed formula obtained by replacing the answer variables $\mathbf{x}$ in q by an answer set $\mathbf{a_i}$, and $\mathcal{K} \models q(\mathbf{a_i})$ means that every model of $\mathcal{K}$ is also model of $q(\mathbf{a_i})$. This corresponds to the well-known certain answers semantics defined in [2, 21]. Given $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ a *DL-Lite* knowledge base and a CQ q, a certain answer to q over $\mathcal{K}$ is an answer that holds in all the models satisfying $\mathcal{K}$.

It is important to note that CQ answering can be reduced to boolean query answering [2]. Namely, given a CQ q with free variables $\mathbf{x}$=$(x_1, ..., x_n)$, an answer set $\mathbf{a}$=$(a_1, ..., a_n)$ is a certain answer for q over $\mathcal{K}$ if the boolean query $q(\mathbf{a})$ obtained by replacing each variable x_i by a_i in $q(\mathbf{x})$, evaluates to true for every model of $\mathcal{K}$.

3 Assertional-based revision of *DL-Lite* knowledge bases

Dynamics of a DL-based knowledge base often concerns the situation where new information should be incorporated while ensuring the consistency of the result (e.g. [51, 53]). Several works recently dealt with revising *DL-Lite* TBox with a terminological information (e.g. [22, 57, 61, 63]) or with an assertional information (e.g. [22, 30, 42]).

In this section, we investigate the revision of *DL-Lite* knowledge bases in the case where priorities are available between assertions in the ABox. We study different forms of the input: An assertion, a positive inclusion axiom or a negative inclusion axiom. We consider a lexicographical strategy where only smallest subsets of assertions should be dropped from the knowledge base in order to restore its consistency and accept the new piece of information. Note that the choice of only dropping information from the ABox is motivated by the fact that in Web applications (such as in Ontology-Based based Access applications) a TBox is often seen as a well-formed and coherent ontology whereas the ABox represents data that are not necessarily reliable and consistent with the ontology. In other words, when the input is a terminological information, the revising process comes down to enrich the ontology while preserving the coherence of the resulting TBox. However, in case of inconsistency, the ABox may be modified in order to take into account the input.

3.1 The notion of conflict

Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be a consistent *DL-Lite* knowledge base. Let us denote by N a new consistent piece of information to be accepted. The presence of this new information may lead to inconsistency according to the content of the TBox and the nature of the input information.

Within the *DL-Lite* language, the new piece of information N may be:

- a membership assertion of the form $A(a)$ or $P(a, b)$,
- a positive inclusion axiom (PI) of the form $B_1 \sqsubseteq B_2$ or
- a negative inclusion axiom (NI) of the form $B_1 \sqsubseteq \neg B_2$.

According to [21], every *DL-Lite* knowledge base $\mathcal{K}$ with only PIs in its TBox is always satisfiable (consequence of Lemma 7 in [21]). Hence, if N is a membership assertion or a PI axiom, there is no inconsistency. However when the TBox $\mathcal{T}$ contains NI axioms then N may have an undesirable interaction with $\mathcal{K}$ which leads to an inconsistency.

We recall that inconsistency in *DL-Lite* is always defined with respect to some ABox, since a TBox may be incoherent but never inconsistent, as stated by Calvanese et al. [22].

Lemma 1 ([22]) *Let $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$ be a DL-Lite knowledge base. If $\mathcal{A}$=∅ then $\mathcal{K}$ is consistent. If $\mathcal{K}$ is inconsistent, then there exists a subset $\mathcal{A}_0 \subseteq \mathcal{A}$ with at most two elements, such that $\langle\mathcal{T},\mathcal{A}_0\rangle$ is inconsistent.*

Let $\mathcal{K}$ be an inconsistent knowledge base, we define the notion of conflict which is a minimal inconsistent subset of $\mathcal{A}$.

Definition 3 Let $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$ be an inconsistent *DL-Lite* knowledge base. A conflict C is a set of membership assertions such that:

- $C \subseteq \mathcal{A}$,
- $\langle\mathcal{T}, C\rangle$ is inconsistent,
- $\forall C', C' \subset C, \left\langle\mathcal{T}, C'\right\rangle$ is consistent.

We denote by $\mathcal{C}(\mathcal{K})$ the collection of conflicts in $\mathcal{K}$. Since $\mathcal{K}$ is assumed to be finite, if $\mathcal{K}$ is inconsistent then $\mathcal{C}(\mathcal{K}) \neq \emptyset$ is also finite. Moreover, note that by Lemma 1 and the fact that $\mathcal{T}$ is coherent, $\forall \mathcal{C} \in \mathcal{C}(\mathcal{K})$, it holds that $|C|$=2.

Example 1 Let $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$ be an inconsistent knowledge base such that
$\mathcal{T}$={ $B_1 \sqsubseteq B_2$, $B_2 \sqsubseteq \neg B_3$} and $\mathcal{A}$={$B_1(a)$, $B_3(a)$, $B_2(b)$, $B_3(b)$, $B_1(c)$}. We have cln ($\mathcal{K}$) ={$B_2 \sqsubseteq \neg B_3$, $B_1 \sqsubseteq \neg B_3$}. Then by Definition 3, $\mathcal{C}(\mathcal{K}) = \{\{B_1(a), B_3(a)\}, \{B_2(b), B_3(b)\}\}$.

3.2 Prioritized *DL-Lite* knowledge base

This section defines the notion of a prioritized *DL-Lite* knowledge base, simply denoted by $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$. We assume that $\mathcal{T}$ is coherent and not stratified. Namely, all elements of $\mathcal{T}$ have the same level of importance. In contrast, the ABox is assumed to be stratified, i.e. partitioned into n strata, $\mathcal{A}$=$\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n$ such that:

- the strata are pairwise disjoint, namely $\forall\mathcal{A}_i, \forall\mathcal{A}_j : \mathcal{A}_i \cap \mathcal{A}_j = \emptyset$,
- the assertions in $\mathcal{A}_i$ have the same level of priority,
- the assertions of $\mathcal{A}_i$ have higher priority than the ones in $\mathcal{A}_j$ when $j > i$.
  Hence assertions in $\mathcal{A}_1$ are the most important ones, while assertions in $\mathcal{A}_n$ are the least important ones.

We first define the lexicographic preference relation between subsets of the ABox as follows.

Definition 4 Let X and X' be two subsets of $\mathcal{A}$. X is strictly preferred to X', denoted by $X <_{lex} X'$, if and only if there exists $\mathcal{I}$, $1 \le i \le n$ such that:

- $|X \cap \mathcal{A}_i| < |X' \cap \mathcal{A}_i|$, and
- $\forall j$, $1 \le j < i$, $|X \cap \mathcal{A}_j| = |X' \cap \mathcal{A}_j|$.

Similarly, X is equally preferred to X', denoted by $X =_{lex} X'$, if and only if $\forall i$, $1 \le i \le n$, $|X \cap \mathcal{A}_i| = |X' \cap \mathcal{A}_i|$. Lastly, X is at least as preferred as X', denoted by $X \le_{lex} X'$, if and only if $X <_{lex} X'$ or $X =_{lex} X'$. The relation $\le_{lex}$ is a total pre-order.

Example 2 Let $\mathcal{A}$ be a stratified ABox, $\mathcal{A}$=$\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ where $\mathcal{A}_1$={$B_1(a)$}, $\mathcal{A}_2$={$B_2(b)$} and $\mathcal{A}_3$={$B_3(a)$, $B_3(b)$}. Let X={$B_3(a)$, $B_3(b)$} and X'={$B_3(a)$, $B_2(b)$} be two subsets of $\mathcal{A}$, we have $X <_{lex} X'$.

3.3 Prioritized removed sets revision of *DL-Lite* knowledge bases

We now investigate the revision of *DL-Lite* knowledge bases according to the nature of the input information. We consider an approach using a lexicographical strategy well-known as ”Prioritized Removed Sets Revision” (PRSR) [6] formerly proposed within a propositional logic setting.

Within the *DL-Lite* framework, in order to restore consistency while keeping new information, the PRSR strategy removes exactly one assertion in each conflict minimizing the number of assertions from $\mathcal{A}_1$, then minimizing the number of assertions in $\mathcal{A}_2$, and so on. Using the lexicographic criterion instead of the set inclusion one, will reduce the set of potential conflicts.

In this paper, we assume that the input N is consistent. This assumption holds when N is a positive or a negative axiom, since a set of axioms may be incoherent, but never inconsistent. When N is an assertional fact, then $\langle \mathcal{T}, N \rangle$ may be inconsistent. In this case, N is simply ignored and the result of revising $\langle \mathcal{T}, \mathcal{A} \rangle$ by N simply leads to the original knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$. Note also that if $\langle \mathcal{T}, N \rangle$ is inconsistent and N is an assertional fact then this means that $\mathcal{T}$ is incoherent, namely it contains some empty concepts.

3.3.1 Revision by a membership assertion

We first consider the case where N is an ABox assertion, which corresponds to the revision by a fact or by an observation. In this case, N is added to a new stratum having the highest and a new priority. Not that if N already belongs to $\mathcal{A}$, then we remove it from its original stratum and add it to the new one. However, in order to avoid heavy notations, we simply write $\mathcal{K} \cup \{N\}$ or $\langle \mathcal{T}, \mathcal{A} \cup \{N\} \rangle$ where $\mathcal{A}$ is a prioritized ABox, to denote the fact that N is added to a new and highest priority stratum of $\mathcal{A}$.

The following definition introduces the concept of prioritized removed sets.

Definition 5 Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ be a consistent stratified knowledge base and N be a membership assertion. A *prioritized removed set*, denoted by X, is a set of membership assertions such that:

- $X \subseteq \mathcal{A}$,
- $\langle \mathcal{T}, (\mathcal{A} \backslash X) \cup \{N\} \rangle$ is consistent,
- $\forall X' \subseteq \mathcal{A}$, if $\left\langle \mathcal{T}, (\mathcal{A} \backslash X') \cup \{N\} \right\rangle$ is consistent then $X \leq_{lex} X'$.

We denote by $\mathcal{PR}(\mathcal{K} \cup \{N\})$ the set of all prioritized removed sets of $\mathcal{K} \cup \{N\}$. If $\mathcal{K} \cup \{N\}$ is consistent then $\mathcal{PR}(\mathcal{K} \cup \{N\}) = \emptyset$. Besides, if $\mathcal{K} \cup \{N\}$ is inconsistent then every conflict C of $\mathcal{K} \cup \{N\}$ contains N. More formally we have the following lemma.

Lemma 2 [2] *Let* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ *be a consistent prioritized knowledge base and* N *be an assertion. If* $\mathcal{K} \cup \{N\}$ *is inconsistent then* $\forall C \in \mathcal{C}(\mathcal{K})$, $N \in C$.

Consequently there exists exactly one prioritized removed set. More formally, we have the following proposition.

[2]all proofs are provided in the Appendix

Proposition 1 *Let $\mathcal{K}$ be a consistent stratified knowledge base and N be a membership assertion. If $\mathcal{K} \cup \{N\}$ is inconsistent then $|\mathcal{PR}(\mathcal{K} \cup \{N\})| = 1$.*

Based on these results, we are able to define the $\circ_{PRSR}$ operator.

Definition 6 Let $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ be a consistent stratified knowledge base and N be a membership assertion. The revised knowledge base $\mathcal{K} \circ_{PRSR} N$ is such that $\mathcal{K} \circ_{PRSR} N = \langle\mathcal{T}, \mathcal{A} \circ_{PRSR} N\rangle$ where $\mathcal{A} \circ_{PRSR} N = (\mathcal{A}\backslash X) \cup \{N\}$ with $\mathcal{PR}(\mathcal{K} \cup \{N\}) = \{X\}$.

Example 3 Let $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ be a consistent stratified knowledge base such that $\mathcal{T}$=$\{B_1 \sqsubseteq B_2, B_2 \sqsubseteq \neg B_3, B_3 \sqsubseteq \neg B_4\}$ and $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ where $\mathcal{A}_1 = \{B_1(a)\}$ $\mathcal{A}_2 = \{B_3(b)\}$, $\mathcal{A}_3 = \{B_4(a)\}$. Let N=$B_3(a)$ then $\mathcal{K} \cup \{N\}$ is inconsistent. By Definition 3, $\mathcal{C}(\mathcal{K} \cup \{N\}) = \{\{B_1(a), B_3(a)\}, \{B_3(a), B_4(a)\}\}$. Hence by Definition 5, $\mathcal{PR}(\mathcal{K} \cup \{N\}) = \{\{B_1(a), B_4(a)\}\}$. Therefore $\mathcal{A} \circ_{PRSR} N = \{B_3(a), B_3(b)\}$ and $\mathcal{A} \circ_{PRSR} N = \mathcal{A}'_1 \cup \mathcal{A}'_2 \cup \mathcal{A}'_3$ where $\mathcal{A}'_1 = \{B_3(a)\}$, $\mathcal{A}'_2 = \{B_3(b)\}$ and $\mathcal{A}'_3 = \emptyset$.

As detailed in Section 5 (Section 5.1) computing the set of conflicts is polynomial. Moreover when the input information is a membership assertion, as stated by Proposition 1 and illustrated in Example 3, there is only one prioritized removed set. Next subsection investigates the case where the input information is a positive or a negative inclusion axiom.

3.3.2 Revision by a positive or a negative axiom

We now consider the case where the input N is a PI axiom or a NI axiom. This new axiom should be added to the TBox and since we gave priority to the TBox over the ABox, the input is kept in the revised knowledge base. In this case, $\mathcal{K} \cup \{N\}$ denotes $\langle\mathcal{T} \cup \{N\}, \mathcal{A}\rangle$. Since $\mathcal{T}$ is considered as non prioritized, then $\mathcal{T} \cup \{N\}$ simply denotes the expansion of $\mathcal{T}$ by N.

Definition 7 Let $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ be a consistent stratified knowledge base and N be a PI or a NI axiom. A prioritized removed set, denoted by X, is a set of assertions such that:

- $X \subseteq \mathcal{A}$,
- $\langle\mathcal{T} \cup \{N\}, (\mathcal{A}\backslash X)\rangle$ is consistent and
- $\forall X' \subseteq \mathcal{A}$, if $\left\langle\mathcal{T} \cup \{N\}, (\mathcal{A}\backslash X')\right\rangle$ is consistent then $X \leq_{lex} X'$.

Let us point out that Definition 7 is similar to Definition 5, except that the input is not added to the ABox but to the TBox. However, the revision process still considers the TBox as a stable knowledge, and hence to restore consistency assertional elements from ABox should be removed. We denote again by $\mathcal{PR}(\mathcal{K} \cup \{N\})$ the set of prioritized removed sets of $\mathcal{K} \cup \{N\}$.

Example 4 Let $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ be a consistent stratified knowledge base such that $\mathcal{T}$=$\{B_1 \sqsubseteq B_2, B_3 \sqsubseteq \neg B_4\}$ and $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ where $\mathcal{A}_1 = \{B_1(a)\}$ $\mathcal{A}_2 = \{B_2(b)\}$, $\mathcal{A}_3 = \{B_3(a), B_3(b)\}$. Let N=$B_2 \sqsubseteq \neg B_3$ then $\mathcal{K} \cup \{N\}$ is inconsistent. $\mathcal{C}(\mathcal{K} \cup \{N\}) = \{\{B_1(a), B_3(a)\}, \{B_2(b), B_3(b)\}\}$. The four possible candidates to be removed are: $X_1 = \{B_1(a), B_2(b)\}$, $X_2 = \{B_1(a), B_3(b)\}$, $X_3 = \{B_3(a), B_2(b)\}$, $X_4 = \{B_3(a), B_3(b)\}$. There is only one prioritized removed set X_4 as illustrated in Table 1.

Table 1 One prioritized removed set

$\mathcal{A}_i$	$\vert X_1 \cap \mathcal{A}_i\vert$	$\vert X_2 \cap \mathcal{A}_i\vert$	$\vert X_3 \cap \mathcal{A}_i\vert$	$\vert X_4 \cap \mathcal{A}_i\vert$
$\mathcal{A}_3$	0	1	1	**2**
$\mathcal{A}_2$	1	0	1	**0**
$\mathcal{A}_1$	1	1	0	**0**

If the stratification of $\mathcal{A}$, now is $\mathcal{A}_1 = \{B_1(a), B_3(a)\}$, $\mathcal{A}_2 = \{B_2(b)\}$, $\mathcal{A}_3 = \{B_3(b)\}$, then there are two prioritized removed sets X_2 and X_4 as illustrated in Table 2.

We have seen that when the input is a membership assertion then there exists exactly one prioritized removed set. However, when the input information is a NI or a PI axiom there may exist one or several prioritized removed sets, as illustrated in the previous example. The first case to consider, which is also the easiest one, is when each conflict intersects two distinct strata. In this case there exists only one prioritized removed set. More formally, the following result holds.

Proposition 2 *If for each $C \in \mathcal{C}(\mathcal{K} \cup \{N\})$ there exists $\mathcal{I}$ and j, $i \neq j$, such that $C \cap \mathcal{A}_i \neq \emptyset$ and $C \cap \mathcal{A}_j \neq \emptyset$ then $|\mathcal{PR}(\mathcal{K} \cup \{N\})| = 1$.*

This situation happens when each stratum is consistent with $\mathcal{T} \cup \{N\}$. In this case, as detailed in Section 5.2, computing the unique prioritized removed set can be performed in polynomial time.

There may be several prioritized removed sets as soon as there are conflicts included in a stratum where each conflict may lead to two prioritized removed sets. Namely, let NC be the number of conflicts such that each one is included in a stratum, the number of prioritized removed sets is bounded by 2^{NC}. In such a case, each prioritized removed set leads to a possible revised knowledge base: $\mathcal{K}_i$=$\langle \mathcal{T} \cup \{N\}, (\mathcal{A} \backslash X_i)\rangle$ with $X_i \in \mathcal{PR}(\mathcal{K} \cup \{N\})$.

Within *DL-Lite* language it is not possible to represent the disjunction of such possible revised knowledge bases. If we want to keep the result of the revision operation expressible in *DL-Lite*, one can define a selection function that selects from $\mathcal{PR}(\mathcal{K} \cup \{N\})$ one or several prioritized removed sets. More formally, a selection function, denoted by f, is defined as follows.

Definition 8 A selection function f is a mapping from $\mathcal{PR}(\mathcal{K} \cup \{N\})$ to $\mathcal{A}$ such that:

- $f(\mathcal{PR}(\mathcal{K} \cup \{N\})) \subseteq \mathcal{A}$
- $\exists X_i \in \mathcal{PR}(\mathcal{K} \cup \{N\})$ such that $X_i \subseteq f(\mathcal{PR}(\mathcal{K} \cup \{N\}))$
- $f(\mathcal{PR}(\mathcal{K} \cup \{N\})) \subseteq \bigcup_{X_i \in \mathcal{PR}(\mathcal{K} \cup \{N\})} X_i$

Table 2 Two prioritized removed sets

$\mathcal{A}_i$	$\vert X_1 \cap \mathcal{A}_i\vert$	$\vert X_2 \cap \mathcal{A}_i\vert$	$\vert X_3 \cap \mathcal{A}_i\vert$	$\vert X_4 \cap \mathcal{A}_i\vert$
$\mathcal{A}_3$	0	**1**	0	**1**
$\mathcal{A}_2$	1	**0**	1	**0**
$\mathcal{A}_1$	1	**1**	1	**1**

The first item in Definition 8 simply states that $f(\mathcal{PR}(\mathcal{K} \cup \{N\}))$ should only contain elements of $\mathcal{A}$. This condition guarantees that the result of revision will be expressible within the *DL-Lite* language. The second item states that at least one prioritized removed set should be in $f(\mathcal{PR}(\mathcal{K} \cup \{N\}))$. This guarantees that $\langle \mathcal{T}, \mathcal{A} \setminus f(\mathcal{PR}(\mathcal{K} \cup \{N\}))\rangle$ is consistent. The last item states that only elements from $\bigcup_{X_i \in \mathcal{PR}(\mathcal{K} \cup \{N\})} X_i$ should be removed and ignored to restore consistency. Hence, elements which are not responsible of conflicts will not be removed.

We now define the revised knowledge base using a selection function as follows.

Definition 9 Let $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A}\rangle$ be a consistent stratified knowledge base and N be a PI or a NI axiom. Let f be a selection function, the revised knowledge base $\mathcal{K} \circ_{PRSR} N$ is such that $\mathcal{K} \circ_{PRSR} N$=$\langle \mathcal{T} \cup \{N\}, \mathcal{A} \circ_{PRSR} N\rangle$ where $\mathcal{A} \circ_{PRSR} N$=$(\mathcal{A} \setminus f(\mathcal{PR}(\mathcal{K} \cup \{N\})))$.

Next subsection presents some examples of selection functions.

3.4 Examples of selection functions

Let us first start with two basic selection functions, denoted simply by f_1 and f_2. The first selection function f_1 consists in taking all prioritized removed sets. More formally,

$$f_1(\mathcal{PR}(\mathcal{K} \cup \{N\})) = \bigcup_{X_i \in \mathcal{PR}(\mathcal{K} \cup \{N\})} X_i,$$

which corresponds to the intersection of all possible revised knowledge bases. In this case $\mathcal{K} \circ_{PRSR} N = \langle \mathcal{T} \cup \{N\}, \mathcal{A} \circ_{PRSR} N\rangle$ where $\mathcal{A} \circ_{PRSR} N$=$\mathcal{A} \setminus f_1(\mathcal{PR}(\mathcal{K} \cup \{N\}))$=$\cap_{i=1}^{n}(\mathcal{A} \setminus X_i)$. This first selection function may be too cautious since it could remove too many assertions and is not in agreement with the minimal change principle.

Another option is to choose a selection function that only picks up one prioritized removed set. More formally,

$$f_2(\mathcal{PR}(\mathcal{K} \cup \{N\})) = X, \text{ for some } X \in \mathcal{PR}(\mathcal{K} \cup \{N\}),$$

which corresponds to the choice of only one revised knowledge base. This option is less cautious than the previous one and captures, in some sense, the existence of a possibility for restoring consistency.

Example 5 Let us consider the knowledge base of Example 2. We have $\mathcal{T}$=$\{B_1 \sqsubseteq B_2, B_3 \sqsubseteq \neg B_4\}$ and $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ where $\mathcal{A}_1 = \{B_1(a), B_3(a)\}$, $\mathcal{A}_2 = \{B_2(b)\}$ and $\mathcal{A}_3 = \{B_3(b)\}$. Let N=$B_2 \sqsubseteq \neg B_3$ be a new piece of information. We have $\mathcal{K} \cup \{N\}$ is inconsistent. The prioritized removed sets are: $X_1 = \{B_1(a), B_3(b)\}$ and $X_2 = \{B_3(a), B_3(b)\}$. We have: $f_1(\mathcal{PR}(\mathcal{K} \cup \{N\})) = \{B_1(a), B_3(b), B_3(a)\}$ and $f_2(\mathcal{PR}(\mathcal{K} \cup \{N\}))$ can be either $\{B_1(a), B_3(b)\}$ or $\{B_3(a), B_3(b)\}$.

The last selection function uses the notion of deductive closure defined as follows.

Definition 10 Let $\mathcal{K}=\langle\mathcal{T},\mathcal{A}\rangle$ be a *DL-Lite* knowledge base. Let $\mathcal{T}_p$ be the set of all positive axioms of $\mathcal{T}$. Let D_C be the set of concepts of $\mathcal{T}$, D_R be the set of roles of $\mathcal{T}$ and D_I be the set of individuals of $\mathcal{A}$. The deductive closure of $\mathcal{A}$ with respect to $\mathcal{T}$ is defined by:

$$\begin{aligned} cl(\mathcal{A}) = & \{A(a) : \langle\mathcal{T}_p, \mathcal{A}\rangle \models A(a)\ with\ a \in D_I\ and\ A \in D_C\} \\ & \cup \\ & \{R(a,b) : \langle\mathcal{T}_p, \mathcal{A}\rangle \models R(a,b)\ with\ a, b \in D_I\ and\ R \in D_R\}. \end{aligned}$$

Using the notion of deductive closure, one can refine the set of prioritized removed sets in which a selection function operates. This new subset, denoted $\mathcal{CPR}(\mathcal{K} \cup \{N\})$, is made by keeping only prioritized removed sets X in $\mathcal{PR}(\mathcal{K}\cup\{N\})$ such that the deductive closure of the set $A \setminus X$ is maximal with respect to the lexicographical criterion.

Definition 11 Let $\mathcal{K}=\langle\mathcal{T},\mathcal{A}\rangle$ be a *DL-Lite* knowledge base and N be the input. Let $\mathcal{PR}(\mathcal{K} \cup \{N\})$ be the set of prioritized removed sets. The set $\mathcal{CPR}(\mathcal{K} \cup \{N\})$ is composed of prioritized removed sets X from $\mathcal{PR}(\mathcal{K} \cup \{N\})$ such that $\nexists Y \in \mathcal{PR}(\mathcal{K} \cup \{N\})$, $|cl(\mathcal{A} \setminus Y)| \geq |cl(\mathcal{A} \setminus X)|$.

Then the last selection function, based on the deductive closure and denoted by f_3, is simply defined by

$$f_3(\mathcal{PR}(\mathcal{K} \cup \{N\})) = \bigcup_{X_i \in \mathcal{CPR}(\mathcal{K}\cup\{N\})} X_i.$$

Clearly, $\mathcal{CPR}(\mathcal{K} \cup \{N\}) \subseteq \mathcal{PR}(\mathcal{K} \cup \{N\})$ then we have $f_2(\mathcal{PR}(\mathcal{K} \cup \{N\})) \subseteq f_3(\mathcal{PR}(\mathcal{K} \cup \{N\})) \subseteq f_1(\mathcal{PR}(\mathcal{K} \cup \{N\}))$. $f_3(\mathcal{PR}(\mathcal{K} \cup \{N\}))$ offers a good compromise between an arbitrary choice of the prioritized removed set to be ignored from the ABox $\mathcal{A}$, and a skeptical choice where all prioritized removed sets are removed from the ABox.

Example 6 From Example 5, one can check that: $cl(\mathcal{A} \setminus X_1) = \{B_3(a), B_2(b)\}$ and $cl(\mathcal{A} \setminus X_2) = \{B_1(a), B_2(a), B_2(b)\}$. Then $\mathcal{CPR}(\mathcal{K} \cup \{N\}) = \{X_2\}$.

In addition to the three above selection functions, we propose a way to directly compute the result of assertional-based revision on the basis of the set of all possible prioritized removed sets. The idea of computing the assertional-based revision is strongly related the notion of universal or skeptical inference that can be defined from $\mathcal{PR}(\mathcal{K} \cup \{N\})$. Namely, we first need to define the set of all possible assertions that can be derived from each $\mathcal{A} \setminus X_i$ with $X_i \in \mathcal{PR}(\mathcal{K} \cup \{N\})$.

More precisely, let D_C be the set of concepts of $\mathcal{T}$, D_R be the set of roles of $\mathcal{T}$ and D_I be the set of individuals of $\mathcal{A}$. Then we define the set of universal assertional consequences, denoted by $UAC(\mathcal{K} \cup \{N\})$ as

$$\begin{aligned} UAC(\mathcal{K} \cup \{N\}) = & \{A(a) : a \in D_I, A \in D_C\ and \\ & \forall X_i \in \mathcal{PR}(\mathcal{K} \cup \{N\}), \langle\mathcal{T}, \mathcal{A} \setminus X_i\rangle \models A(a)\} \\ & \cup \\ & \{R(a,b) : a \in D_I, b \in D_I, R \in D_R\ and \\ & \forall X_i \in \mathcal{PR}(\mathcal{K} \cup \{N\}), \langle\mathcal{T}, \mathcal{A} \setminus X_i\rangle \models R(a,b)\}. \end{aligned}$$

Example 7 Let us consider $\mathcal{T}$={$A \sqsubseteq B, C \sqsubseteq B$} and $\mathcal{A} = \mathcal{A}_1$ where $\mathcal{A}_1$={$B(b)$, $A(a)$, $C(a)$}. Let N=$A \sqsubseteq \neg C$ be a new piece of information. We have $\mathcal{K} \cup \{N\}$ is inconsistent. The two possible prioritized removed sets that can be computed are: $X_1 = \{A(a)\}$ and $X_2 = \{C(a)\}$.
One can check that $UAC(\mathcal{K} \cup \{N\})$={$B(a)$, $B(b)$}.

In fact $UAC(\mathcal{K} \cup \{N\})$ contains all assertions of the form $A(a)$ or $R(a, b)$ (where a, b are individuals, A is a concept and R is a role) that can be derived by ignoring each possible prioritized removed set.

3.5 Multiple revision

In the previous sections, it is assumed that the input information is only composed of a single element: An assertional fact, a positive axiom or a negative axiom. This section briefly discusses the case where the input contains more than one element. This problem is known as multiple revision and has been addressed for instance in [29, 34] in a propositional setting.

Let $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ be a *DL-Lite* prioritized knowledge base. Let us start with the situation where the input, simply denoted again by N, is a set of assertional facts. If $\langle\mathcal{T}, N\rangle$ is consistent, then our approach can be applied straightforwardly. The definition of prioritized removed set is exactly the same. Definitions 5 to 11 can be used as it is, except that N is a set of assertional facts instead of a single one. The same holds for Lemma 2 as well as Propositions 1–2.

Example 8 Let us consider $\mathcal{T}$={$A \sqsubseteq \neg B$} and $\mathcal{A}$=$\mathcal{A}_1 \cup \mathcal{A}_2$ where $\mathcal{A}_1$={$A(a)$, $B(c)$} and $\mathcal{A}_2$={$B(b)$}. Let N={$A(b)$, $B(a)$} where $\langle\mathcal{T}, N\rangle$ is consistent. Now, $\langle\mathcal{T}, \mathcal{A} \cup N\rangle$ is inconsistent. There only exists one prioritized removed set: X_1={$A(a)$, $B(b)$} and $\mathcal{A} \circ_{PRSR} N$={$A(b)$, $B(a)$, $B(c)$}.

Now assume that $\langle\mathcal{T}, N\rangle$ is inconsistent. In this case, if we still consider that $\mathcal{T}$ as a stable knowledge, then the input cannot be wholly accepted. In this case, the prioritized removed set will both contains elements from $\mathcal{A}$ and also from N, with elements of N being preferred to all elements of $\mathcal{A}$. Definition 5 needs a small adaptation asfollows:

Definition 12 Let $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ be a consistent stratified knowledge base and N be a set of membership assertion. A *prioritized removed set*, denoted by X, is a set of membership assertions such that:

- $X \subseteq \mathcal{A} \cup N$,
- $\langle\mathcal{T}, (\mathcal{A} \cup N) \setminus X\rangle$ is consistent,
- $\forall X' \subseteq \mathcal{A} \cup N$, if $\langle\mathcal{T}, (\mathcal{A} \cup N) \setminus X'\rangle$ is consistent then $X \leq_{lex} X'$.

Note that $\mathcal{A} \cup N$ is a new prioritized ABox, when elements of N are put in a new important stratum. Namely, let $\mathcal{A}$=$\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$ be a prioritized ABox. Then $\mathcal{A} \cup N$=$\mathcal{A}'_1 \cup \ldots \cup \mathcal{A}'_{n+1}$ where $\mathcal{A}'_1 = N$, and $\mathcal{A}'_i = \mathcal{A}_{i-1}$ for $i = 1, ..., n+1$.

The remaining definitions are valid, however Lemma 1 and Proposition 1 do not hold as it is shown by the following counter-example.

Example 9 Let us consider $\mathcal{T}$={$A \sqsubseteq \neg B$} and $\mathcal{A}$=$\mathcal{A}_1 \cup \mathcal{A}_2$ where $\mathcal{A}_1$={$A(a)$} and $\mathcal{A}_2$={$B(c)$}. Let N={$A(b)$, $B(b)$, $B(a)$} where $\langle\mathcal{T}, N\rangle$ is inconsistent. We have $\langle\mathcal{T}, N \cup \mathcal{A}\rangle$

is also inconsistent. The conflicts are: C_1={$A(a)$, $B(a)$} and C_2={$A(b)$, $B(b)$}. The two prioritized removed sets are: X_1={$A(a)$, $A(b)$} and X_2={$A(a)$, $B(b)$}.

One can check that there exist more than one prioritized removed set which both contain elements from $\mathcal{A}$ and N.

As said previously, when the input N is a set of PI axioms or NI axioms, we assume that $\mathcal{T} \cup N$ is coherent, since the TBox of the knowledge base is assumed to be stable. Of course $\langle \mathcal{T} \cup N, \mathcal{A} \rangle$ may be inconsistent. In this case PRSR behaves in the same way as simple revision by a single input. In both cases (set of assertion or axioms), the most noticeable difference is that the number of conflicts may be higher and by consequence the size of prioritized removed sets may be higher.

Lastly, if the input contains both membership assertions and PI axioms or NI axioms, then this comes down to revise the *DL-Lite* prioritized knowledge base $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ with another knowledge base $N = \langle \mathcal{T}', \mathcal{A}' \rangle$. One way to achieve such revision is to apply PRSR on $\langle \mathcal{T} \cup \mathcal{T}', \mathcal{A} \cup \mathcal{A}' \rangle$.

Example 10 Let us consider $\mathcal{T}$={$A \sqsubseteq \neg B$} and $\mathcal{A}$=$\mathcal{A}_1 \cup \mathcal{A}_2$ where $\mathcal{A}_1$={$A(a)$} and $\mathcal{A}_2$={$B(b)$}. Let N=$\langle \mathcal{T}' = \{C \sqsubseteq A\}, \mathcal{A}' = \{C(a), A(b)\} \rangle$. We have $\langle \mathcal{T} \cup \mathcal{T}', \mathcal{A} \cup \mathcal{A}' \rangle$ is inconsistent. There exists only one conflict C={$A(b)$, $B(b)$} and hence one prioritized removed set X={$B(b)$}.

4 Logical properties

In this section we go a step further in the characterization of Prioritized Removed Sets Revision by presenting logical properties of the proposed operators through a set of postulates.

As mentioned in the introduction, the AGM postulates [1] have been expressed to characterize belief revision in a propositional logic setting. Flouris at al. have studied which logics are AGM-compliant, that is, DLs where the revision operation satisfies AGM postulates [25–27]. Indeed, the problem is that AGM postulates are defined for belief sets, i.e. deductively closed sets of formulas, possibly infinite. Qi et al. [53] focused on revising a finite representation of belief sets. They used a semantic reformulation of AGM postulates, done by Katsuno and Mendelzon [41], to extend it to DLs knowledge bases. However, as pointed out in [22], known model-based approaches of revision are not expressible in *DL-Lite*. AGM postulates are defined for belief sets, however efficient implementation and computational tractability require finite representations. Moreover, cognitive realism stems from finite structures [38] since infinite structures are cognitively inaccessible. Revision within the framework of DLs, particularly, *DL-Lite*, requires belief bases, i.e. finite sets of formulas. Postulates have been proposed for characterizing belief bases revision in a propositional logic setting [28, 37].

In order to give logical properties of PRSR operators, we first rephrase Hansson's postulates within the *DL-Lite* framework. We then analyze to what extent our operators satisfy these postulates.

4.1 Hansson's postulates reformulated

Let $\mathcal{K}$, $\mathcal{K}'$ be two *DL-Lite* knowledge bases, N and M be either membership assertions or positive or a negative axioms, $\circ$ be a revision operator. $\mathcal{K} + N$ denotes the non closing expansion, i.e. $\mathcal{K} + N = \mathcal{K} \cup \{N\}$. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base. When N is a membership assertion $\mathcal{K}+N = \langle \mathcal{T}, \mathcal{A} \cup \{N\} \rangle$ and when N is a positive or a negative axiom $\mathcal{K} + N = \langle \mathcal{T} \cup \{N\}, \mathcal{A} \rangle$. We rephrase the Hansson's postulates as follows.

Success	$N \in \mathcal{K} \circ N$
Inclusion	$\mathcal{K} \circ N \subseteq \mathcal{K} + N$.
Consistency	$\mathcal{K} \circ N$ is consistent.
Vacuity	If $\mathcal{K} \cup \{N\}$ is consistent then $\mathcal{K} \circ N = \mathcal{K} + N$.
Pre-expansion	$(\mathcal{K} + N) \circ N = \mathcal{K} \circ N$.
Internal exchange	If $N, M \in \mathcal{K}$ then $\mathcal{K} \circ N = \mathcal{K} \circ M$.
Core retainment	If $M \in \mathcal{K}$ and $M \notin \mathcal{K} \circ N$ then there exists $\mathcal{K}'$ such that $\mathcal{K}' \subseteq \mathcal{K} + N$ and $\mathcal{K}'$ is consistent but $\mathcal{K}' \cup \{M\} is$ inconsistent.
Relevance	If $M \in \mathcal{K}$ and $M \notin \mathcal{K} \circ N$ then there exists $\mathcal{K}'$ such that $\mathcal{K} \circ N \subseteq \mathcal{K}' \subseteq \mathcal{K} + N$, and $\mathcal{K}'$is consistent but $K' \cup \{M\}$ is inconsistent.

Success and *Consistency* express the basic principles of revision. *Inclusion* states that the union of the initial knowledge bases is the upper bound of any revision operation. *Vacuity* says that if new information is consistent with the knowledge base then the result of revision equals the non closing expansion. *Pre-expansion* states that expanding first by an assertion does not change the result of revision by the same assertion. *Internal exchange* says that revising by two different assertions from the knowledge base does not change the result of revision. In fact, since $\mathcal{K}$ is assumed to be consistent, then this postulate is always satisfied, since if $M \in \mathcal{K}$ then $\mathcal{K} \circ M$=$\mathcal{K}$. *Core-retainment* and *Relevance* express the intuition that nothing is removed from the original knowledge bases unless its removal contributes in some way to make the result consistent.

4.2 Prioritized removed sets revision: logical properties

We now present the logical properties of PRSR operators. We recall that, when the input is a membership assertion, there is only one removed set.

Proposition 3 *Let $\mathcal{K}$ be a consistent stratified DL-Lite knowledge base and N be a membership assertion. Then the revision operator $\circ_{PRSR}$ satisfies Success, Inclusion,* Consistency*, Vacuity, Pre-expansion, Internal exchange, Core retainment and Relevance.*

This proposition states that PRSR with a membership assertion as input satisfies all postulates. The situation is slightly different when N is a PI or a NI axiom. In this case, the definition of the revision operation requires a selection function, as stated in Definition 9.

Proposition 4 *Let $\mathcal{K}$ be a consistent stratified DL-Lite knowledge base. If N is a PI or a NI axiom then for any selection function, the revision operator $\circ_{PRSR}$ satisfies Success, Inclusion, Consistency, Vacuity, Pre-expansion, Internal exchange, Core retainment but does not satisfy Relevance.*

Relevance requires the existence of only one prioritized removed set which is the case when N is a membership assertion. However, when N is a PI or a NI axiom, in general, there may exist several prioritized removed sets.

5 Computing the revision operation outcome

As stated before, when trying to revise a *DL-Lite* knowledge base by a membership assertion, a PI axiom or a NI axiom, we want to withdraw only ABox assertions in order to restore consistency, i.e. prioritized removed sets will only contain elements from the ABox.

From the computational point of view, we have to distinguish several cases depending on the nature of the input N and the content of the knowledge base.

First of all, if the TBox $\mathcal{T}$ only contains PI axioms, and if the input N is a PI axiom or a membership assertion, no inconsistency can occur, so the revision operation PRSR trivially becomes a simple union. Among the remaining cases, we distinguish two different situations:

1. N is a membership assertion: The computation of conflicts and the overall revision algorithm is a very simple task, thanks to Proposition 1, as detailed in Section 5.2.1.
2. N is a PI axiom or a NI axiom : This is the most complicated case, as several prioritized removed sets may exist. Moreover, we will see that this case has to be splitted into two subcases (see Section 5.2.2). Whatever case we consider, we first need to compute the conflicts of $\mathcal{K} \cup \{N\}$.

In what follows, we use the following notations: $\mathcal{K}'$=$\langle\mathcal{T}', \mathcal{A}'\rangle$=$\mathcal{K} \cup \{N\}$. Thus, if N is a PI or NI axiom we have $\mathcal{T}'$=$\mathcal{T} \cup \{N\}$ and $\mathcal{A}'$=$\mathcal{A}$, and if N is an ABox assertion, we have $\mathcal{T}'$=$\mathcal{T}$ and $\mathcal{A}'$=$\mathcal{A} \cup \{N\}$.

5.1 Computing the conflicts

This step follows from the algorithm given in [21] for checking the consistency of a *DL-Lite* knowledge base. The main difference is that in [21] the aim is to check whether a *DL-Lite* knowledge base is consistent or not. Here, we have to perform one step further, as we need to enumerate all assertional pairs involved in conflicts. Hence, we need to slightly adapt the algorithm.

Computing $\mathcal{C}(\mathcal{K} \cup \{N\})$ first requires to obtain the negative closure $cln\left(\mathcal{T}'\right)$, using the rules recalled in the refresher on *DL-lite* logic in Section 2. We suppose that this is performed by a NEGCLOSURE function. Then the computation of the conflicts proceeds with the evaluation over $\mathcal{A}'$ of each NI axiom in $cln\left(\mathcal{T}'\right)$ in order to exhibit whether $\mathcal{A}'$ contains pairs of assertions that contradict the NI axioms. Intuitively, for each $C_1 \sqsubseteq \neg C_2$ (resp. $R_1 \sqsubseteq \neg R_2$) belonging to $cln\left(\mathcal{T}'\right)$, the evaluation of $C_1 \sqsubseteq \neg C_2$ (resp. $R_1 \sqsubseteq \neg R_2$) over the $\mathcal{A}'$ simply amounts to return all $(C_1(x), C_2(x))$ (resp. $(R_1(x, y), R_2(x, y))$) such that $C_1(x)$ and $C_2(x)$ (resp. $R_1(x, y)$ and $R_2(x, y)$) belong to $\mathcal{A}'$. Note then $C_1(x)$ (*resp.* $C_2(x)$) may be a basic concept assertion, or a role assertion of the form $R(x, y)$ if $C_1 = \exists R$ (*resp.* $C_2 = \exists R$) or $R(y, x)$ if $C_1 = \exists R^-$ (*resp.* $C_2 = \exists R^-$). The result of the evaluation of a NI axiom is a collection of sets containing two elements, or one element if N is a membership assertion. Algorithm 1 describes the algorithm of the function COMPUTECONFLICTS, which computes $\mathcal{C}(\mathcal{K} \cup \{N\})$.

The set $\mathcal{C}(\mathcal{K}')$ stores the conflicts. The first step of the algorithm consists in the computation of the negative closure of $\mathcal{T}'$. Then, for each NI axiom $X \sqsubseteq \neg Y$ of $cln\left(\mathcal{T}'\right)$ the algorithm looks for the existence of a contradiction in the ABox. This is done by checking whether $\langle X \sqsubseteq \neg Y, \{\alpha_t, \alpha_j\}\rangle$ is consistent or not. Note that this step can be performed by a boolean query expressed from $X \sqsubseteq \neg Y$ to look whether $\{\alpha_t, \alpha_j\}$ contradicts the query, or not. If the ABox is consistent with $X \sqsubseteq \neg Y$, then the result of the query is empty set.

Algorithm 1 COMPUTECONFLICTS($\mathcal{K}$)

function COMPUTECONFLICTS($\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, N)
  $\mathcal{K}' = \langle \mathcal{T}', \mathcal{A}' \rangle \leftarrow \mathcal{K} \cup \{N\}$
  $\mathcal{C}(\mathcal{K}') \leftarrow \emptyset$
  $cln\left(\mathcal{T}'\right) \leftarrow$ NEGCLOSURE($\mathcal{T}'$)
  for all $X \sqsubseteq \neg Y \in cln\left(\mathcal{T}'\right)$ **do**
    for all $\left\{\alpha_t, \alpha_j\right\} \subseteq \mathcal{A}'$ **do**
      if $\left\langle X \sqsubseteq \neg Y, \left\{\alpha_t, \alpha_j\right\}\right\rangle$ is inconsistent **then**
        $\mathcal{C}(\mathcal{K}') \leftarrow \mathcal{C}(\mathcal{K}') \cup \left\{\left\{\alpha_t, \alpha_j\right\}\right\}$
  Return $\mathcal{C}(\mathcal{K}')$

It is important to note that if N is a membership assertion, then in each conflict $\{\alpha_t, \alpha_j\}$ either α_t or α_j belongs to $\mathcal{A}$ (but not both), and that either α_t or α_j is equal to N (but not both). This special case is detailed in the next subsection.

5.2 Computing the PRSR outcome

5.2.1 Revision by an assertion

When the input N is a membership assertion (namely a fact), then there exists only one prioritized removed set, and the priorities are not involved. The computation of this single prioritized removed set amounts to pick up in each conflict the membership assertion which is different from new information N. One can easily check that every conflict $\left\{\alpha_t, \alpha_j\right\}$ that contradicts a NI axiom is of the form $\{\alpha, N\}$ where $\alpha \in \mathcal{A}$. This means that there exists exactly one prioritized removed set. Hence, in this case the prioritized removed set computation can be performed in polynomial time, namely, when returning from the call to COMPUTECONFLICTS, the only prioritized removed set is $\bigcup_{c_i \in \mathcal{C}(\mathcal{K} \cup \{N\})} (c_i \setminus \{N\})$.

Algorithm 2 describes the algorithm of the function COMPUTEPRSR1 as a special case of Algorithm 1. It computes directly the single prioritized removed set when revising by a membership assertion.

Algorithm 2 COMPUTEPRSR1($\mathcal{K}$, N)

function COMPUTEPRSR1($\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, $N = A(a)$ or $N = R(a, b)$)
  $R \leftarrow \emptyset$
  $cln\,(\mathcal{T}) \leftarrow$NEGCLOSURE($\mathcal{T}$)
  for all $X \sqsubseteq \neg Y \in cln\,(\mathcal{T})$ **do**
    for all $\alpha \in \mathcal{A}$ **do**
      if $\langle X \sqsubseteq \neg Y, \{\alpha, N\} \rangle$ is inconsistent **then**
        $R \leftarrow R \cup \{\alpha\}$
return R

5.2.2 Revision by An axiom

Now, we detail the case where N is a PI or a NI axiom. According to Definition 7, the computation of $\mathcal{PR}(\mathcal{K} \cup \{N\})$ starts with the computation of $\mathcal{PR}((\mathcal{T} \cup \{N\}) \cup \mathcal{A}_1)$, followed by the computation of $\mathcal{PR}((\mathcal{T} \cup \{N\}) \cup (\mathcal{A}_1 \cup \mathcal{A}_2))$, and so on where the $\mathcal{A}_i$'s are the different strata in the assertional base. A prioritized removed set is formed by picking up in each conflict the element having the lowest priority level. However, according to the form of conflicts, two situations hold, as pointed out in Section 3.3. The first one is when *each conflict* involves two elements having different levels of priority. In this case, Proposition 2 ensures that there exists only one prioritized removed set. We provide the algorithm COMPUTEPRSR2 which computes this single prioritized removed set $PR \in \mathcal{PR}(\mathcal{K} \cup \{N\})$.

Algorithm 3 COMPUTEPRSR2

function COMPUTEPRSR2 ($\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, N)
  $Res \leftarrow \mathcal{A}_1$
  $PR \leftarrow \emptyset$
  $\mathcal{C} \leftarrow$COMPUTECONFLICTS$(\mathcal{K}, N)$
  for $i \leftarrow 2, n$ **do**
    for all $\alpha \in \mathcal{A}_i$ **do**
      if $\exists C \in \mathcal{C}, Res \cap C \neq \emptyset$ **and** $\alpha \in C$ **then**
        $PR \leftarrow PR \cup \{\alpha\}$
        $\mathcal{A}_i \leftarrow \mathcal{A}_i \setminus \{\alpha\}$
    $Res \leftarrow Res \cup \mathcal{A}_i$
  Return PR

The algorithm COMPUTEPRSR2 proceeds from the first layer to all the other less preferred ones and selects the assertions that conflict with the assertions of the current layer. Here we increment from a layer to another in order to ensure the minimality of the prioritized removed set with respect to the lexicographic ordering. Note that this algorithm is based on the inconsistency checking algorithm, and thus, its computational complexity is polynomial (see Section 6).

Now, we detail the second case where there exists at least a conflict involving two elements having the same priority level. In such a situation there exists several prioritized removed sets, as pointed out in Section 3.3.2. To compute them, we use the hitting set notion [59] and we adapt it to the stratified structure of the knowledge base.

Prioritized removed sets are not necessarily minimal with respect to cardinality, but they are minimal with respect to lexicographic ordering ($\leq_{lex}$ for short). So, a naive algorithm for computing $\mathcal{PR}(\mathcal{K} \cup \{N\})$ could be : (i) Compute the kernels of $\mathcal{C}(\mathcal{K} \cup \{N\})$. (ii) keep only minimal ones with respect to $\leq_{lex}$. However, we can improve this algorithm. As we said before, a prioritized removed set is computed from a layer to another. The idea of the enhancement of the algorithm is as follows: Compute conflicts in the first layer, i.e. in $\langle \mathcal{T} \cup \{N\}, \mathcal{A}_1 \rangle$. Then, build the hitting set tree on this collection of conflicts. This tree allows for the computation of the kernels of $\langle \mathcal{T} \cup \{N\}, \mathcal{A}_1 \rangle$, which are minimal with respect to $\leq_{lex}$.

From these kernels, we continue the construction of the tree using conflicts in $\langle \mathcal{T} \cup \{N\}, \{\mathcal{A}_1 \cup \mathcal{A}_2\} \rangle$ if they exist, and so on until reaching a fixed point where no conflict will be generated. Then, the kernels of the final hitting set tree — i.e. those built using the conflicts in $\langle \mathcal{T} \cup \{N\}, \{\mathcal{A}_1 \cup \mathcal{A}_2 \cup ... \cup \mathcal{A}_n\} \rangle$ — which are minimal with respect to $\leq_{lex}$ are the prioritized removed sets. Algorithm 4 describes the algorithm of the function COMPUTEPRSR3, which computes $\mathcal{PR}(\mathcal{K} \cup \{N\})$.

In this algorithm, the function HS$(\mathcal{C})$ takes as input the conflicts computed in each strata (if they exist) and builds the corresponding hitting sets tree (TREE) using the algorithm

Algorithm 4 COMPUTEPRSR3

```
function COMPUTEPRSR3 (𝒦=⟨𝒯, 𝒜⟩ , N)
    𝒯′ ← 𝒯 ∪ {N}, 𝒦′ = ⟨𝒯′, 𝒜⟩
    cln (𝒯′) ← NEGCLOSURE(𝒯′)
    𝒫ℛ(𝒦′) ← ∅ , 𝒞 ← ∅, TREE← ∅, i ← 1
    while i ≤ n do
        for all X ⊑ ¬Y ∈ cln (𝒯′) do
            for all (α, β) s.t. α ∈ 𝒜1, β ∈ 𝒜1 ∪ ... ∪ 𝒜i do
                if ⟨X ⊑ ¬Y, {α, β}⟩ is inconsistent then
                    𝒞 ← 𝒞 ∪ {α, β}
        TREE← TREE.ADDFROMLEXKERNEL(HS(𝒞))
        𝒞 ← ∅,
        i ← i + 1
    𝒫ℛ(𝒦′) ← LEXKERNEL(TREE)
    return 𝒫ℛ(𝒦′)
```

presented in [59, 62]. From one layer to another, we resume the construction of (TREE) from its current kernels minimal with respect to $\leq_{lex}$. Namely, the function ADDFROMLEXKERNEL(HS($\mathcal{C}$)) build the hitting set tree out of a collection of conflicts $\mathcal{C}$, starting from the branches of the current TREE which are minimal with respect to $\leq_{lex}$. Finally $\mathcal{PR}(\mathcal{K} \cup \{N\})$ corresponds to the kernels of TREE obtained using function LEXKERNEL(TREE) which are minimal with respect to $\leq_{lex}$. Note that COMPUTEPRSR3 is a generalization of COMPUTEPRSR2, since when all conflicts involve elements from distinct layers, then the final tree will only contain one prioritized removed set. The following example illustrates this algorithm.

Example 11 Consider $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$, with $\mathcal{T}$=$\{A \sqsubseteq B, C \sqsubseteq B\}$ and $\mathcal{A}$=$\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4$ where $\mathcal{A}_1$=$\{A(a), D(a)\}$, $\mathcal{A}_2$=$\{C(a), B(b)\}$, $\mathcal{A}_3$=$\{D(b)\}$ and $\mathcal{A}_4$=$\{D(c), C(c)\}$. We want to revise $\mathcal{K}$ with N=$B \sqsubseteq \neg D$. Then, we have $cln\,(\mathcal{T} \cup \{B \sqsubseteq \neg D\})$=$\{B \sqsubseteq \neg D, A \sqsubseteq \neg D, C \sqsubseteq \neg D\}$. The set of conflicts obtained from $\langle cln\,(\mathcal{T}'), \mathcal{A}_1\rangle$ is $\{\{A(a), D(a)\}\}$. The HS tree built by calling HS($\{\{A(a), D(a)\}\}$) will contains two branches labeled respectively by $A(a)$ and $D(a)$ which are kernels minimal with respect to $\leq_{lex}$ ($\leq_{lex}$-kernel). We go on with $\langle cln\,(\mathcal{T}'), \mathcal{A}_1 \cup \mathcal{A}_2\rangle$ where $\{C(a), D(a)\}$ is a newly identified conflict. We resume the construction of the tree from its current $\leq_{lex}$-kernel branches labeled by $A(a)$ and $D(a)$, and we obtain three HS-tree branches: $\{A(a), C(a)\}$, $\{A(a), D(a)\}$ and $D(a)$, where only $D(a)$ is a $\leq_{lex}$-kernel. Now, we step to the next strata, that is, we use $\langle cln\,(\mathcal{T}'), \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3\rangle$. This produces a new conflict $\{B(b), D(b)\}$ and we continue the construction of the Tree from $D(a)$. We potentially obtain $\{D(a), D(b)\}$ and $\{D(a), B(b)\}$ as new prioritized removed sets, but only $\{D(a), D(b)\}$ is a $\leq_{lex}$ kernel. Finally, we identify a new conflict $\{D(c), C(c)\}$ from $\langle cln\,(\mathcal{T}'), \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4\rangle$. We continue the construction of the tree from the branch labeled by $\{D(a), D(b)\}$. We obtain two other branches labeled respectively by $\{D(a), D(b), C(c)\}$ and $\{D(a), D(b), D(c)\}$ which are two $\leq_{lex}$ kernels. Hence, $\mathcal{PR}(\mathcal{K} \cup \{N\})$=$\{\{D(a), D(b), C(c)\}, \{D(a), D(b), D(c)\}\}$. The construction of the HS tree is illustrated in Fig. 1.

The problem of revising a prioritized *DL-Lite* knowledge base is closely related to the problem of computing knowledge base repairs. Indeed, in this paper the presence of inconsistency is due to the addition of a new sure piece of information. Inconsistency may also

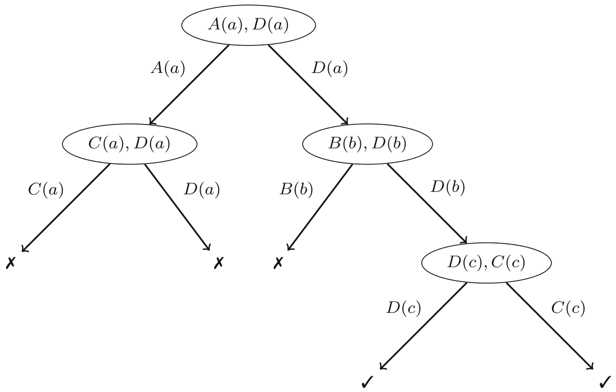

Fig. 1 Construction of the HS tree

be the result of concatenating several consistent *DL-Lite* knowledge bases issued from different sources. The result is then a prioritized inconsistent *DL-Lite* knowledge base where priorities may simply reflect the reliability levels associated with each source.

Handling inconsistency comes down to compute different repairs of the prioritized inconsistent knowledge base. Clearly, the Prioritized Removed Sets Revision (PRSR) approach can be easily adapted to compute repairs. Algorithm 1 can be used as it is. Algorithm 2 in the case of multiple sources information is useful for incoherent *DL-Lite* knowledge base. Namely, if A is an incoherent concept, Algorithm 2 allows us to remove all individuals belonging to A. Algorithm 2 is only useful for this case. Algorithm 3 basically can be used as is. It suffices to remove the parameter N from lines 1 to 7, and $\mathcal{K}$ is assumed to be possibly inconsistent and issued from different sources. Similarly, Algorithm 4 can be basically used as is by removing the parameter N from line 1 and 2.

6 Computational complexity analysis

This section studies the computational complexity of PRSR operations for *DL-Lite* knowledge bases. Traditionally, in the knowledge revision community, the problem considered for computational complexity analysis ([23, 47]) is the inference problem. The parameters of this problem are: A knowledge base K, an input sentence α, and any sentence ϕ. Given an input of the problem, the question is: Can we infer the sentence ϕ from the revised knowledge base $K * \alpha$?

However, this problem is too general in the case of revision of $DL\text{-}Lite_R$ knowledge bases. The $DL\text{-}Lite_R$ logic aims at applications which typically use knowledge bases consisting in very large evolving ABoxes, while relying on TBoxes which are of a smaller size (compared to the ABox). The main usage of such bases is querying, i.e. information extraction, and instance checking, i.e. checking whether an individual is a member of a concept, or whether two individuals are members of a role. Thus, our complexity analysis of PRSR revision operation will focus on the problems of instance checking and conjunctive query answering. More precisely, these problems are defined as follows.

Problem: INSTANCE CHECKING($\circ_{PRSR}$).
Input: A $DL\text{-}Lite_R$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, an input N (which is either an assertion, a positive inclusion axiom, or a negative inclusion axiom), a concept

C (resp. a role R) from the TBox and an individual a (resp. two individuals a, b) from the ABox.

Question: Does $\mathcal{K} \circ_{PRSR} N \models C(a)$ (resp. $\mathcal{K} \circ_{PRSR} N \models R(a, b)$) hold ?

Problem: CONJUNCTIVE QUERY ANSWERING($\circ_{PRSR}$).
Input: A $DL\text{-}Lite_R$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, an input N, a conjunctive query $q(\mathbf{x})$, and a tuple of constants $\mathbf{a}$ of $\mathcal{K}$.
Question: Does $\mathcal{K} \circ_{PRSR} N \models q(\mathbf{a})$ hold ?

The expression of the conjunctive query answering problem may seem a little bit odd. Obviously, the real problem, would be to compute the result of the query evaluation over $\mathcal{K} \circ_{PRSR} N$, denoted by $Ans(q, \mathcal{K} \circ_{PRSR} N)$. Our expression of the problem represents the equivalent decision problem, as previously proposed in [21], an reminded in Section 2.4.

As for the description of the computation of the revision outcome, we distinguish three different cases, depending on the nature of the input N and on the stratification of the ABox $\mathcal{A}$ of the knowledge base:

Case 1: The input is a membership assertion;
Case 2: The input is an inclusion axiom and *each conflict involves two assertions from different strata* (denoted hereafter DS-case);
Case 3: The input is an inclusion axiom and there exist *conflicts that involve two assertions from the same stratum* (denoted hereafter SS-case).

Finally, for each problem INSTANCE CHECKING($\circ_{PRSR}$) and CONJUNCTIVE QUERY ANSWERING($\circ_{PRSR}$), and each of the preceding cases, we consider *data complexity*, that is, complexity depending on the size of the ABox, and *combined complexity*, that is, complexity depending on the size of the TBox and the ABox.

For the needs of the proofs of the following results, we recall the complexity results of $DL\text{-}Lite_R$. The Instance Checking problem is in AC^0 for data complexity [2], and is in NLOGSPACE for combined complexity [2][3]. The Conjunctive Query Answering problem is AC^0 for data complexity [2], and in NLOGSPACE for combined complexity[2][3].

Before detailing the complexity results and their proofs, we give a synthetic view of these results in Table 3.

Note that the proofs of computational complexity of query answering include the analysis of the computational complexity of the different algorithms presented in Section 5 for computing the resulted base revision.

6.1 Case 1: *N* is a membership assertion

The computation of the only prioritized removed set can be performed by the COMPUTEPRSR1 function described in Algorithm 2.

Proposition 5 *The complexity of the* COMPUTEPRSR1 *function is quadratic in the size of the ABox (data complexity) and cubic in the size of the TBox and the ABox (combined complexity).*

[3]the combined complexity result is not provided explicitly in this article, but it stems from the complexity class of the satisfiability problem in *DL-Lite*.

Table 3 Complexity results of $\circ_{PRSR}$ operators

N	Instance checking($\circ$)		Conjunctive query answering($\circ$)	
	Data	Combined	Data	Combined
Membership assertion(1)	Quadratic	Cubic	Quadratic	Cubic
Inclusion axiom (DS)(2)	Quadratic	Cubic	Quadratic	Cubic
Inclusion axiom (SS)(3)	Θ_2^p	Θ_2^p	Θ_2^p	Θ_2^p

Proposition 6 CONJUNCTIVE QUERY ANSWERING($\circ_{PRSR}$) *(resp.* INSTANCE CHECKING($\circ_{PRSR}$)*) is quadratic for data complexity and cubic for combined complexity.*

6.2 Case 2: *N* is an inclusion axiom, DS-case

In this case, we consider that each conflict involves two assertions located in different strata. In this situation, thanks to Proposition 2, we know that there is only one prioritized removed set, which can be computed by the function COMPUTEPRSR2 described in Algorithm 3.

Proposition 7 *The complexity of the* COMPUTEPRSR2 *function is quadratic in the size of the ABox (data complexity) and cubic in the size of the TBox and the ABox (combined complexity).*

This result allows us to establish the following complexity results.

Proposition 8 CONJUNCTIVE QUERY ANSWERING*($\circ_{PRSR}$) (resp.* INSTANCE CHECKING($\circ_{PRSR}$)*) is quadratic for data complexity and cubic for combined complexity.*

6.3 Case 3: *N* is an inclusion axiom, SS-case

In this case, we consider that there exist some conflicts which involve two assertions located in the same strata. Intuitively this means that in order to restore consistency, we can choose the assertion to remove from the conflict, as they have the same level of priority. Thus, this multiplies by two the number of prioritized removed sets for each choice.

Proposition 9 CONJUNCTIVE QUERY ANSWERING*($\circ_{PRSR}$) (resp.* INSTANCE CHECKING*($\circ_{PRSR}$)) is in Θ_2^p for both data and combined complexity.*

7 Related works

In [22], Calvanese et al. study the problem of knowledge base *evolution* in *DL-Lite*. Under the word evolution, they encompass both revision and update operations. Note that the update focuses on the changes of the actual state whereas revision focuses on the integration of new information [61]. In this paper, we focus on revision.

The part of this article dedicated to so-called "formula-based approaches" is closely related to our work. They define several operators which perform revision of a knowledge base expressed in *DL-Lite* at a syntactical level.

The difference is that in [22] they develop two operators whose strategy is to non-derministically choose some maximal consistent subset. The first one, called $BoldEvol$, starts with the input, and incrementally and non-deterministically adds as many formulas as possible from the closure of the knowledge base. The algorithm for computing such set is polynomial. However, in the case where the input is a set of membership assertions, they give a result similar to our operator, namely, the result only gives one maximal consistent subset, which corresponds to Proposition 1.

The selected maximal subset is a subset of the *consequences* of the knowledge base, which is very different from our point of view. Prioritized Removed Sets Revision relies only on the explicit content of the knowledge base. The resulting knowledge base will not contain formulas which are not present in the original knowledge base. By only working with explicitly given information, we follow Hansson's point of view [38].

Following this line, extensions of belief bases revision to Description logics have been proposed, however these approaches differ from ours in several aspects.

Within the general framework of Description Logics, in [52] the authors extend kernel-based revision [35] for revising flat terminologies. Our approach is very different since we deal with knowledge bases which are prioritized and expressed in a lightweight Description Logic. Furthermore, our revision operators do not modify the TBox but revise the prioritized ABox according to a lexicographical strategy.

In [33], the authors focus on $\mathcal{SHOIN}$ Description Logic, they extend kernel revision and semi-revision operators [36] to $\mathcal{SHOIN}$ knowledge bases. Moreover, they propose an algorithm for revision stemming from the computation of kernels. This algorithm shares several common points with our algorithm for the computation of prioritized removed sets. What they call justification of the inconsistency is very similar to our notion of conflict. But in their case, the generation of conflicts has a higher computational cost than in our case, as they work with $\mathcal{SHOIN}$ logic. In order to lower this extra-complexity, they rely on an optimized version of the Pellet consistency checker which uses properties of the $\mathcal{SHOIN}$ logic, allowing them to define an incremental version of their consistency checking tableau algorithm.

In [60], the authors propose another extension of kernel-based revision and semi-revision operators to Description Logics, namely external kernel revision and semi-revision with weak success. Once again, their logical framework is richer than our, since they consider $\mathcal{SHOIN}$ and $\mathcal{SHIf}$ logics in order to capture all the *OWL-DL* and *OWL-Lite* languages. Our revision operators can be viewed as restrictions of the operator they define under the name *kernel revision without negation*. The restrictions are : (i) our knowledge bases are prioritized and expressed in DL-Lite; (ii) the minimality of the result of the incision function is defined in terms of lexicographic criterion in our case.

Following another idea, the authors in [54], extend weakening-based revision to $\mathcal{ALC}$ knowledge bases. Instead of removing conflicting assertions, the proposed revision operators weaken terminological axioms or assertions by adding exceptions which drop individuals responsible of the conflicts. Furthermore, this weakening-based revision is generalized to stratified knowledge bases. Our revision operators differ from this approach since our prioritized knowledge bases are expressed in *DL-lite*. Moreover, the spirit is different since PRSR removes conflicting assertions according to a lexicographical strategy.

Our revision approach is also related to another important problem that often appears in the Ontology-Based Data Access. It is the problem of answering (complex or simple) queries addressed to an inconsistent knowledge base expressed in *DL-Lite*. Recently several works [17, 19, 32, 45] have been proposed to deal with such a problem. Those works are especially inspired by the approaches proposed in databases which stem from the notion of database repair to answering queries raised to inconsistent databases. A repair of a database contradicting a set of integrity constraints is a database obtained by applying a minimal set of changes in order to restore consistency. The notion of database repair has been extended to ABox repair for DL knowledge bases.

Concerning the computational complexity analysis, it has been shown in [17, 45] that some inconsistency-tolerant inferences in *DL-Lite* are coNP-complete. Namely deciding whether a query or an instance checking holds from all repairs of a *DL-Lite* knowledge base is in coNP for Data complexity (Theorem 3 in [45]). As detailed in Section 6, we have shown that using the PRSR approach we obtain lexicographic-based inference relations with a slightly similar computational complexity.

8 Conclusions and future work

In this paper, we investigated the problem of revising prioritized *DL-Lite* knowledge bases where the ABox is stratified. We considered several forms of incorporated information, more precisely, when the input is a membership assertion, a positive or negative inclusion axiom. According to the form of the input we proposed a family of operators, Prioritized Removed Sets (PRSR) operators, stemming form a lexicographical strategy for removing some assertions, namely the prioritized removed sets, in order to restore consistency.

When the input is a membership assertion, the revision process leads to a unique revised knowledge base. However, when the input is a positive or negative inclusion axiom, the revision process may lead to several possible revised knowledge bases. In this case, we defined selection functions in order to keep the result within the *DL-Lite* language and we gave some concrete PRSR operators with examples of selection functions.

We studied the logical properties of PRSR operators through Hansson's postulates rephrased within the *DL-Lite* framework.

From a computational point of view, we first proposed an algorithm for pinpointing inconsistencies, then according to the nature of the input, we proposed algorithms, some of them using the notion of hitting set, for computing the prioritized removed sets. Finally, we conducted a complexity analysis of the proposed algorithms and identified the cases where PRSR can be achieved in polynomial time.

In this paper, the proposed postulates capture the general behavior of revision operators but they do not capture the specificity of prioritized revision. In a future work, a study will be dedicated to the formulation of new postulates taking the priorities into account.

It is well known that nonmonotonic reasoning and revision are considered as the two sides of the same coin [46]. A further investigation should focus on inconsistency-tolerant inference relations defined from the family of PRSR revision operators.

A future work will focus on a deeper study of the computational complexity of the PRSR operators that could be further refined, in particular, the completeness with respect to class should be checked.

Finally, we plan to study the merging of *DL-Lite* knowledge bases when data are provided by several equally reliable sources. In a near future, we will investigate the extension of Removed Sets Fusion [39], defined in a propositional setting, to the merging of *DL-Lite* knowledge bases.

Acknowledgments We thank Nadia Creignou for helpful comments on complexity analysis. We would like to thank anonymous reviewers for their careful reading and valuables comments and suggestions to improve the paper. This work has received support from the french Agence Nationale de la Recherche, ASPIQ project reference ANR-12-BS02-0003.

Appendix

Proofs

Lemma 2 *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a consistent prioritized knowledge base and N be an assertion. If $\mathcal{K} \cup \{N\}$ is inconsistent then $\forall C \in \mathcal{C}(\mathcal{K})$ it holds that $N \in C$.*

Proof The proof is immediate. If $\mathcal{K} \cup \{N\}$ is inconsistent then $\mathcal{C}(\mathcal{K} \cup \{N\}) \neq \emptyset$. This means that there exists at least a conflict $C = (\alpha, \beta) \in \mathcal{C}(\mathcal{K} \cup \{N\})$ (Recall that $|C|$=2). Let $\mathcal{C}$ be a conflict of $\mathcal{K} \cup \{N\}$. Suppose that $N \notin C$. This means that $\alpha \in \mathcal{A}$ and $\beta \in \mathcal{A}$. This is a contradiction since $\mathcal{K}$ is assumed to be consistent, namely $\mathcal{C}(\mathcal{K})$=$\emptyset$. □

Proposition 1 *Let $\mathcal{K}$ be a consistent stratified knowledge base and N be a membership assertion. If $\mathcal{K} \cup \{N\}$ is inconsistent then $|\mathcal{PR}(\mathcal{K} \cup \{N\})| = 1$.*

Proof Suppose that there are two prioritized removed sets X and X' such that $X \neq X'$. By Definition 5, $X \subseteq \mathcal{A}$, $X' \subseteq \mathcal{A}$ and $X =_{lex} X'$. Since $(\mathcal{T} \cup \{N\}) \cup (\mathcal{A} \backslash X)$ and $(\mathcal{T} \cup \{N\}) \cup (\mathcal{A} \backslash X')$ are consistent, we have $\forall C \in \mathcal{C}(\mathcal{K} \cup \{N\})$ on one hand $C \cap X \neq \emptyset$ and $|C \cap X| = 1$ and on the other hand $C \cap X' \neq \emptyset$ and $|C \cap X'| = 1$. Moreover, since N is a single assertion, by Lemma 2, $|C \cap N| = 1$. Therefore there are three elements in C namely N, one element of X and one element of X'. Hence, this contradicts Lemma 1 that states that $|C \cap \mathcal{A}| \leq 2$. □

Proposition 2 *If for each $C \in \mathcal{C}(\mathcal{K} \cup \{N\})$ there exists i and j, $i \neq j$, such that $C \cap \mathcal{A}_i \neq \emptyset$ and $C \cap \mathcal{A}_j \neq \emptyset$ then $|\mathcal{PR}(\mathcal{K} \cup \{N\})| = 1$.*

Proof Suppose there are two prioritized removed sets, X and X' and $X \neq X'$. By Definition 7, $X \subseteq \mathcal{A}$, $X' \subseteq \mathcal{A}$, and $X =_{lex} X'$. Since $(\mathcal{T} \cup \{N\}) \cup (\mathcal{A} \backslash X)$ and $(\mathcal{T} \cup \{N\}) \cup \left(\mathcal{A} \backslash X'\right)$ are consistent, $\forall C \in \mathcal{C}(\mathcal{K} \cup \{N\})$ we have $C \cap X \neq \emptyset$ and $C \cap X \neq \emptyset$. If $|C \cap X|$=2 (resp. $|C \cap X'|$=2) then X (resp. X') is not a prioritized removed set, since C is a minimal inconsistent subset with two elements by Lemma 1. If $|C \cap X|$=1 and $|C \cap X'|$=1 two cases hold. If $C \cap X \neq C \cap X'$ since there exists i and j, $i \neq j$, such that $C \cap \mathcal{A}_i \neq \emptyset$ and $C \cap \mathcal{A}_j \neq \emptyset$ it contradicts $X =_{lex} X'$. If $C \cap X$=$C \cap X'$, since C intersects two strata, and $|C \cap X|$=$|C \cap X'|$=1 then X=X' which contradicts the hypothesis. □

Proposition 3 *Let $\mathcal{K}$ be a consistent stratified DL-Lite knowledge base and N be a membership assertion. Then the revision operator $\circ_{PRSR}$ satisfies Success, Inclusion, Consistency, Vacuity, Pre-expansion, Internal exchange, Core retainment and Relevance.*

Proof Since N is a membership assertion, $\mathcal{K} \cup \{N\} = \langle \mathcal{T}, \mathcal{A} \cup \{N\} \rangle$. By Definition 6, $\mathcal{K} \circ_{PRSR} N = \langle \mathcal{T}, \mathcal{A} \circ_{PRSR} N \rangle$ with $\mathcal{A} \circ_{PRSR} N = (\mathcal{A} \backslash X) \cup \{N\}$ and the postulates *Success*, *Inclusion*, *Consistency* are satisfied.

Vacuity: If $\mathcal{K} \cup \{N\}$ is consistent, then $\mathcal{PR}(\mathcal{K} \cup \{N\})$=$\emptyset$ and $\mathcal{A} \circ_{PRSR} N = \mathcal{A} \cup \{N\}$, therefore the postulate holds.

Pre-expansion: $(\mathcal{A} \cup \{N\}) \circ_{PRSR} N = ((\mathcal{A} \cup \{N\}) \backslash X) \cup \{N\} = (\mathcal{A} \backslash X) \cup \{N\}$, therefore the postulate is satisfied.

Internal exchange: If $N, M \in \mathcal{A}$, $\mathcal{A} \cup \{M\}$=$\mathcal{A} \cup \{N\}$=$\mathcal{A}$ and $\mathcal{PR}(\mathcal{K} \cup \{N\})$=$\mathcal{P}R(\mathcal{K} \cup \{M\})$=$\emptyset$, therefore the postulate is satisfied.

Core retainment: The case where $M \in \mathcal{T}$ is impossible since the $\circ_{PRSR}$ operator may only modify the ABox. When M is a membership assertion, if $M \in \mathcal{K}$ and $M \notin \mathcal{K} \circ_{PRSR} N$ then there exists X such that $M \in X$ and $X \in \mathcal{PR}(\mathcal{K} \cup \{N\})$. Let $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A} \backslash X \rangle$, we have $\mathcal{K}' \subseteq \mathcal{K} \cup \{N\}$ et $\mathcal{K}'$ is consistent but $\mathcal{K}' \cup \{M\}$ is inconsistent, therefore the postulate is satisfied.

Relevance: Since the postulate *Core retainment* is satisfied, and by Proposition 1 we have $|\mathcal{PR}(\mathcal{K} \cup \{N\})|$=1, so $\mathcal{K} \circ_{PRSR} N \subseteq \mathcal{K}'$ and thus the postulate holds.

□

Proposition 4 *Let $\mathcal{K}$ be a consistent stratified DL-Lite knowledge base. If N is a PI or a NI axiom then for any selection function, the revision operator $\circ_{PRSR}$ satisfies Success, Inclusion, Consistency, Vacuity, Pre-expansion, Internal exchange, Core retainment but does not satisfy Relevance.*

Proof Since N is a positive or a negative axiom, $\mathcal{K} \cup \{N\} = \langle \mathcal{T} \cup \{N\}, \mathcal{A} \rangle$. By Definition 9, $\mathcal{K} \circ_{PRSR} N$=$\langle \mathcal{T} \cup \{N\}, \mathcal{A} \circ_{PRSR} N \rangle$ with $\mathcal{A} \circ_{PRSR} N = (\mathcal{A} \backslash f(\mathcal{R}(\mathcal{K} \cup \{N\})))$ and the postulates *Success*, *Inclusion*, *Consistency* are satisfied.

Vacuity: If $\mathcal{K} \cup \{N\}$ is consistent, $\mathcal{PR}(\mathcal{K} \cup \{N\}) = \emptyset$ and $\mathcal{A} \circ_{PRSR} N = \mathcal{A}$, therefore the postulate holds.

Pre-expansion: $(\mathcal{K} \cup \{N\}) \circ_{PRSR} N, (\langle \mathcal{T} \cup \{N\}, \mathcal{A} \rangle) \circ_{PRSR} N = \langle \mathcal{T} \cup \{N\}, \mathcal{A} \circ_{PRSR} N \rangle$, therefore the postulate is satisfied.

Internal exchange: If $N, M \in \mathcal{T}$, $\mathcal{T} \cup \{M\} = \mathcal{T} \cup \{N\} = \mathcal{T}$ and $\mathcal{PR}(\mathcal{K} \cup \{N\}) = \mathcal{PR}(\mathcal{K} \cup \{M\}) = \emptyset$, therefore the postulate is satisfied.

Core retainment: The case where $M \in \mathcal{T}$ is impossible since the $\circ_{PRSR}$ operator may only modify the ABox. When M is a membership assertion, if $M \in \mathcal{K}$ and $M \notin \mathcal{K} \circ_{PRSR} N$, then for any selection function used for defining $\circ_{PRSR}$, there exists $X \in \mathcal{PR}(\mathcal{K} \cup \{N\})$ such that $M \in X$ and $X \subseteq f(\mathcal{R}(\mathcal{K} \cup \{N\}))$ by Definition 8. Let $\mathcal{K}' = \langle \mathcal{T} \cup \{N\}, \mathcal{A} \backslash X \rangle$, we have $\mathcal{K}' \subseteq \mathcal{K} \cup \{N\}$ and $\mathcal{K}'$ is consistent but $\mathcal{K}' \cup \{M\}$ is inconsistent, therefore the postulate is satisfied.

Relevance: Since the postulate *Core retainment* is satisfied, there exists $\mathcal{K}' = \langle \mathcal{T} \cup \{N\}, \mathcal{A} \backslash X \rangle$ consistent. Since there may exist several prioritized removed sets, let X and X' be two prioritized removed sets such that $X' \neq X$, suppose that $f(\mathcal{PR}(\mathcal{K} \cup \{N\})) = X'$, we have $\mathcal{K} \circ_{PRSR} N = \langle \mathcal{T} \cup \{N\}, \mathcal{A} \backslash X' \rangle$ therefore $\mathcal{K} \circ_{PRSR} N \not\subseteq \mathcal{K}'$, therefore the postulate *Relevance* is not satisfied. We now give a counter-example where $\mathcal{K}$ and N come from Example 4. Let $M = B_3(b)$, $X = \{B_3(a), B_3(b)\}$ and $X' = \{B_1(a), B_3(b)\}$ be two prioritized removed sets, suppose that $f(\mathcal{PR}(\mathcal{K} \cup \{N\})) = X'$ we have $\mathcal{A} \circ_{PRSR} N = \{B_3(a), B_2(b)\}$ and $\mathcal{A} \backslash X = \{B_1(a), B_2(b)\}$.

□

Proposition 5 *The complexity of the* COMPUTEPRSR1 *function is quadratic in the size of the ABox (data complexity) and cubic in the size of the TBox and the ABox (combined complexity).*

Proof In line 3 of Algorithm 2, the computation of $cln\ (\mathcal{T})$ can be performed in $O(n^3)$ time in the size of the TBox. The two loops in lines 5 and 6 can be performed in $O(n^2)$ time in the size of the ABox. Thus, the data complexity of the the algorithm is quadratic, and the combined complexity is cubic. □

Proposition 6 CONJUNCTIVE QUERY ANSWERING($\circ_{PRSR}$) *(resp.* INSTANCE CHECKING($\circ_{PRSR}$)*) is quadratic for data complexity and cubic for combined complexity.*

Proof Since in this case there is only one prioritized removed set, answering to this question can be performed by the following algorithm :

$S \leftarrow$COMPUTEPRSR1($\mathcal{K}$, N)
if $\mathcal{K} \setminus \{S\} \models q(\mathbf{a})$ **then**
  return true
else
  return false

Data complexity: COMPUTEPRSR1 is quadratic in the size of the ABox. Checking wether $\mathcal{K} \setminus \{S\} \models q(\mathbf{a})$ is AC^0in the size of the ABox. Thus, the preceding algorithm is quadratic in the size of the ABox.

Combined complexity: COMPUTEPRSR1 is cubic in the size of the whole knowledge base. Checking wether $\mathcal{K} \setminus \{S\} \models q(\mathbf{a})$ is in NLOGSPACE. Thus the problem is cubic in the combined size of the ABox ad the TBox. The proof of instance checking follows similarly.

□

Proposition 7 *The complexity of the* COMPUTEPRSR2 *function is quadratic in the size of the ABox (data complexity) and cubic in the size of the TBox and the ABox (combined complexity).*

Proof As already mentioned, the computation of the conflicts of $\mathcal{K} \cup N$ using the function COMPUTECONFLICTS can be performed in $O(n^3)$ in the size of the TBox and in $O(n^2)$ in the size of the ABox (the proof is similar to the proof of Proposition 5).

Moreover, the loops in line 5 and line 6 will perform a full examination of the $|\mathcal{A}|$ assertions in the ABox, and the test in line 7 can, in the worst case, fully browse the $|\mathcal{A}|$ assertions. Thus, the worst case data complexity of the two loops and the test is in $O(n^2)$. These loops and the test do not involve the TBox, so they have no impact on the combined complexity. □

Proposition 8 CONJUNCTIVE QUERY ANSWERING($\circ_{PRSR}$) *(resp.* INSTANCE CHECKING($\circ_{PRSR}$)*) is quadratic for data complexity and cubic for combined complexity.*

Proof Since in this case there is only one removed set, answering to this question can be performed by the following algorithm :

$S \leftarrow$COMPUTEPRSR2($\mathcal{K}$, N)
if $\mathcal{K} \setminus \{S\} \models q(\mathbf{a})$ **then**
  return true
else
  return false

Data complexity: ComputePRSR2 is quadratic in the size of the ABox. Checking wether $\mathcal{K} \backslash \{S\} \models q(\mathbf{a})$ is LogSpace in the size of the ABox. Thus, the preceding algorithm is quadratic in the size of the ABox.

Combined complexity: ComputePRSR2 is cubic in the size of the whole knowledge base. Checking wether $\mathcal{K} \setminus \{S\} \models q(\mathbf{a})$ is in NLogSpace. Thus the problem is cubic for combined complexity. The proof of instance checking follows similarly. □

Proposition 9 Conjunctive Query Answering($\circ_{PRSR}$) *(resp.* Instance Checking($\circ_{PRSR}$)*) is in* Θ_2^p *for both data and combined complexity.*

Proof We consider the question : does $\mathcal{K} \circ_{PRSR} N \models q(\mathbf{a})$? We can use the following algorithm:

k_{prio} ←ComputeSizePrio($\mathcal{K}$, N)
repeat
    gess R
    if $|R| = k_{prio}$ and $(\mathcal{K} \cup \{N\}) \setminus R$ is consistent **then**
        if $(\mathcal{K} \cup \{N\}) \setminus \{R\} \models q(\mathbf{a})$ **then**
            return true
until all removed sets have been computed
return false

We denote by ComputeSizePrio($\mathcal{K}$, N) the algorithm which computes the size of prioritized removed sets. This computation requires $\log |\mathcal{A}|$ calls to an NP-Oracle, namely HITTING SET ([31], p.222) on the collection $\mathcal{C}(\mathcal{K} \cup \{N\})$.

The main loop will be performed an exponential number of times (on the total number of formulas in $\mathcal{K}$), and : (i) and checking wether $(\mathcal{K} \cup \{N\}) \setminus R$ is consistent is in NLogSpace, (ii) checking wether $(\mathcal{K} \cup \{N\}) \setminus \{R\} \not\models q(\mathbf{a})$ is in $AC^0$for data complexity, NLogSpace for combined complexity. Thus, the most expensive part of the algorithm is the computation of k_{prio} and the problem is in Θ_2^p for both data and combined complexity. The proof of Instance Checking($\circ_{PRSR}$) follows similarly. □

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. Symb. Log. **50**(2), 510–530 (1985)
2. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. Artif. Intell. Res. **36**, 1–69 (2009)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation and Applications. 2nd. Cambridge University Press, New York (2010)
4. Baader, F., Hollunder, B.: Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. J. Autom. Reason. **15**(1), 41–68 (1995)
5. Baader, F., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The description logic handbook: Theory, implementation, and applications. Cambridge University Press (2003)
6. Benferhat, S., Ben-Naim, J., Papini, O., Würbel, E.: An answer set programming encoding of prioritized removed sets revision: application to gis. Appl. Intell. **32**(1), 60–87 (2010)
7. Benferhat, S., Bouraoui, Z., Papini, O., Wùrbel, E.: Assertional-based prioritized removed sets revision of DL-LiteR knowledge bases. In: Proceeding of ECAI′14, pp. 967–968 (2014)
8. Benferhat, S., Bouraoui, Z., Papini, O., Würbel, E.: Assertional-based removed sets revision of DL-Lite$_R$ knowledge bases. In: Proceedings of ISAIM′14 (2014)

9. Benferhat, S., Bouraoui, Z., Papini, O., Würbel, E.: A prioritized assertional-based revision for DL-Lite knowledge bases Fermé, E., Leite, J. (eds.) Proceedings of JELIA′14, volume 8761 of Lecture Notes in Computer Science, pages 442–456. Springer (2014)
10. Benferhat, S., Bouraoui, Z., Tabia, K.: On the revision of prioritized DL-Lite knowledge bases. In: Straccia, U., Calì, A. (eds.) Proceedings of SUM′14, volume 8720 of Lecture Notes in Computer Science, pages 22–36. Springer (2014)
11. Benferhat, S., Bouraoui, Z., Tabia, K.: How to select one preferred assertional-based repair from inconsistent and prioritized DL-Lite knowledge bases?. In: Yang, Q., Wooldridge, M. (eds.) Proceedings of IJCAI′15, pages 1450–1456. AAAI Press (2015)
12. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H.: Inconsistency management and prioritized syntax-based entailment. In: Bajcsy, R. (ed.) Proceedings of IJCAI′93, pages 640–647. Morgan Kaufmann (1993)
13. Benferhat, S., Dubois, D., Lang, J., Prade, H., Saffiotti, A., Smets, P.: A general approach for inconsistency handling and merging information in prioritized knowledge bases. In: Cohn, A.G., Schubert, L.K. (eds.) Proceedings of KR′98, pages 466–477. Morgan Kaufmann (1998). Shapiro, S.C.
14. Benferhat, S., Dubois, D., Prade, H.: How to infer from inconsisent beliefs without revising? In Proceedings of IJCAI′95, pages 1449–1457. Morgan Kaufmann (1995)
15. Benferhat, S., Dubois, D., Prade, H., Williams, M.-A.: A practical approach to revising prioritized knowledge bases. Studia Logica **70**(1), 105–130 (2002)
16. Benferhat, S., Dubois, D., Prade, H., Williams, M.-A.: A framework for iterated belief revision using possibilistic counterparts to jeffrey's rule. Fundamenta Informaticae **99**(2), 147–168 (2010)
17. Bienvenu, M.: Inconsistency-tolerant conjunctive query answering for simple ontologies. In: Kazakov, Y., Lembo, D., Wolter, F. (eds.) Proceedings of DL′12, volume 846 of CEUR Workshop Proceedings. CEUR-WS.org (2012)
18. Bienvenu, M., Bourgaux, C., Goasdoué, F.: Querying inconsistent description logic knowledge bases under preferred repair semantics. In Proceedings of AAAI′14, 996–1002 (2014)
19. Bienvenu, M., Rosati, R.: New inconsistency-tolerant semantics for robust ontology-based data access. In: Eiter, T., Glimm, B., Kazakov, Y., Krötzsch, M. (eds.) Proceedings of DL′13, volume 1014 of CEUR Workshop Proceedings, pages 53–64. CEUR-WS.org (2013)
20. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: DL-Lite: Tractable description logics for ontologies. In Proceedings of AAAI′2005, pages 602–607. AAAI Press / The MIT Press (2005)
21. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The dl-lite family. J. Autom. Reason. **39**(3), 385–429 (2007)
22. Calvanese, D., Kharlamov, E., Nutt, W., Zheleznyakov, D.: Evolution of DL-Lite knowledge bases. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) Proceedings of ISWC′10, volume 6496 of LNCS, pages 112–128. Springer (2010)
23. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artif. Intell. **57**(2-3), 227–270 (1992)
24. Flouris, G., Huang, Z., Pan, J.Z., Plexousakis, D., Wache, H.: Inconsistencies, negations and changes in ontologies. In: Proceedings of AAAI′06, pages 1295–1300. AAAI Press (2006)
25. Flouris, G., Plexousakis, D., Antoniou, G.: Generalizing the agm postulates: preliminary results and applications. In: Delgrande, J.P., Schaub, T. (eds.) Proceedings of NMR′04, 171–179 (2004)
26. Flouris, G., Plexousakis, D., Antoniou, G.: On applying the agm theory to dls and owl. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) Proceedings of ISWC′05, volume 3729 of Lecture Notes in Computer Science, pages 216–231. Springer (2005)
27. Flouris, G., Plexousakis, D., Antoniou, G.: On generalizing the agm postulates (2006)
28. Fuhrmann, A.: An essay on contraction. CSLI Publications, Stanford. California (1997)
29. Fuhrmann, A., Hansson, S.O.: A survey of multiple contractions. J. Log. Lang. Inf. **3**(1), 39–75 (1994)
30. Gao, S., Qi, G., Wang, H.: A new operator for abox revision in DL-Lite. In: Hoffmann, J., Selman, B. (eds.) Proceedings of AAAI′12. AAAI Press (2012)
31. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
32. Giacomo, G.D., Lenzerini, M., Poggi, A., Rosati, R.: On the approximation of instance level update and erasure in description logics. In: Proceedings of AAAI′07, pages 403–408. AAAI Press (2007)
33. Halaschek-wiener, C., Katz, Y., Parsia, B.: Belief base revision for expressive description logics. In: Proceedings of of Workshop on OWL Experiences and Directions (2006)
34. Hansson, S.O.: A dyadic representation of belief, 89–121 (1992)
35. Hansson, S.O.: Kernel contraction. J. Symb. Log. **59**, 845–859 (1994)
36. Hansson, S.O.: Semi-revision. J. Appl. Non-classical Log. **7**, 151–175 (1997)

37. Hansson, S.O.: Revision of belief sets and belief bases. Handb. Defeasible Reason. Uncertainty Manag. Syst. **3**, 17–75 (1998)
38. Hansson, S.O.: Specified meet contraction. Erkenntnis **69**(1), 31–54 (2008)
39. Hué, J., Würbel, E., Papini, O.: Removed sets fusion: Performing off the shelf. In: Proceedings of ECAI′08, pp. 94–98 (2008)
40. Ji, Q., Gao, Z., Huang, Z.: Conflict resolution in partially ordered OWL DL ontologies. In: Proceedings of ECAI′2014, pp. 471–476 (2014)
41. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Artif. Intell. **52**, 263–294 (1991)
42. Kharlamov, E., Zheleznyakov, D.: Understanding inexpressibility of model-based abox evolution in DL-Lite. In: Barceló, P., Tannen, V. (eds.) Proceedings of AMW′11, volume 749 of CEUR Workshop Proceedings. CEUR-WS.org (2011)
43. Kharlamov, E., Zheleznyakov, D., Calvanese, D.: Capturing model-based ontology evolution at the instance level: The case of DL-Lite. J. Comput. Syst. Sci. **79**(6), 835–872 (2013)
44. Lee, K., Meyer, T.: A classification of ontology modification. In: Webb, G., Yu, X. (eds.) AI 2004: Advances in Artificial Intelligence, volume 3339 of Lecture Notes in Computer Science, pages 248–258. Springer Berlin Heidelberg, p. 2005
45. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Hitzler, P., Lukasiewicz, T. (eds.) Proceedings of RR′10, volume 6333 of LNCS, pages 103–117. Springer (2010)
46. Makinson, D., Gärdenfors, P.: Relations between the logic of theory change and nonmonotonic logic. In: Fuhrmann, A., Morreau, M. (eds.) The Logic of Theory Change, volume 465 of Lecture Notes in Computer Science, pages 185–205. Springer (1989)
47. Nebel, B.: Base revision operations and schemes: Semantics, representation and complexity. In: Proceedings of ECAI′94, pp. 341–345 (1994)
48. Nonfjall, H., Larsen, H.L.: Detection of potential inconsistencies in knowledge bases. Int. J. Intell. Syst. **7**(2), 81–96 (1992)
49. Papini, O.: A complete revision function in propositional calculus. In: Proceedings of ECAI′92, pp. 339–343 (1992)
50. Poggi, A., Lembo, D., Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. Data Semant. **10**, 133–173 (2008)
51. Qi, G., Du, J.: Model-based revision operators for terminologies in description logics. In: Boutilier, C. (ed.) Proceedings of IJCAI′09, 891–897 (2009)
52. Qi, G., Haase, P., Huang, Z., Ji, Q., Pan, J.Z., Vȯlker, J.: A kernel revision operator for terminologies - algorithms and evaluation. In: Proceedings of ISWC′08, pp. 419–434 (2008)
53. Qi, G., Liu, W., Bell, D.A.: Knowledge base revision in description logics. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) Proceedings of JELIA′06, volume 4160 of LNCS, pages 386–398. Springer (2006)
54. Qi, G., Liu, W., Bell, D.A.: A revision-based approach to handling inconsistency in description logics. Artif. Intell. Rev. **26**(1-2), 115–128 (2006)
55. Qi, G., Liu, W., Bell, D.A.: A revision-based approach to handling inconsistency in description logics. Artif. Intell. Rev. **26**(1-2), 115–128 (2006)
56. Qi, G., Pan, J.Z.: A stratfication-based approach for inconsistency handling in description logics. In: Proceedings of IWOD′07, p. 83 (2007)
57. Qi, G., Wang, Z., Wang, K., Fu, X., Zhuang, Z.: Approximating model-based abox revision in DL-Lite: Theory and practice. In: Bonet, B., Koenig, S. (eds.) Proceedings of AAAI′15, pages 254–260. AAAI Press (2015)
58. Qi, G., Yang, F.: A survey of revision approaches in description logics. In: Baader, F., Lutz, C., Motik, B. (eds.) Proceedings of DL′08, volume 353 of CEUR Workshop Proceedings. CEUR-WS.org (2008)
59. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. **32**(1), 57–95 (1987)
60. Ribeiro, M.M., Wassermann, R.: Base revision in description logics — preliminary results. In: Proceedings of WOD′07, Innsbruck, Austria (2007)
61. Wang, Z., Wang, K., Topor, R.W.: A new approach to knowledge base revision in DL-Lite. In: Fox, M., Poole, D. (eds.) Proceedings of AAAI′10. AAAI Press (2010)
62. Wilkerson, R.W., Greiner, R., Smith, B.A.: A correction to the algorithm in reiter's theory of diagnosis. Artif. Intell. **41**, 79–88 (1989)
63. Zhuang, Z., Wang, Z., Wang, K., Qi, G.: Contraction and revision over DL-Lite tboxes. In: Proceedings of AAAI′14, pp. 1149–1156 (2014)