A Prioritized Assertional-Based Revision for DL-Lite Knowledge Bases

Salem Benferhat[1], Zied Bouraoui[1], Odile Papini[2], and Eric Würbel[3]

[1] CRIL-CNRS UMR 8188, Univ Artois, France
{benferhat,bouraoui}@cril.univ-artois.fr
[2] LSIS-CNRS UMR 7296, Univ Aix Marseille, France
odile.papini@univ-amu.fr
[3] LSIS-CNRS UMR 7296, Univ Toulon, France
wurbel@univ-tln.fr

Abstract. *DL-Lite* is a powerful and tractable family of description logics. One of the fundamental issue in this area is the evolution or revision of knowledge bases. To this end, many approaches are recently developed for revising flat *DL-Lite* knowledge bases. This paper investigates "Prioritized Removed Sets Revision" (PRSR) in *DL-Lite* framework where the assertions or data are prioritized (for instance in case where the data are provided by multiple sources having different reliability levels). PRSR approach is based on inconsistency minimization in order to restore consistency where the minimality refers to the lexicographic criterion and not to the set inclusion one. We study different forms of incorporated information: an assertion, a positive inclusion axiom or a negative inclusion axiom. We show that under some conditions PRSR can be achieved in polynomial time. We give logical properties of the proposed operators in terms of satisfaction of Hansson's postulates rephrased in *DL-Lite* framework. We finally show how to use the notion of hitting sets for computing prioritized removed sets.

1 Introduction

In the last years, there has been an increasing use of ontologies in many application areas. Description Logics (DLs) have been recognized as a powerful formalism for both representing and reasoning about ontologies. 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 the assertional facts (i.e. individuals or constants) that instantiate terminological knowledge. Recently, a lot of attention was given to *DL-Lite* [8], a family of lightweight DLs specifically tailored for applications that use huge volumes of data, like Web applications, for which query answering is the most important reasoning task. *DL-Lite* guarantees a very low computational complexity of the reasoning process.

DLs knowledge base evolution gave rise to an increasing interest (e.g. [10,17]) and often concerns the situation where new information should be incorporated, while ensuring the consistency of the result. Such problem is well-known as a belief revision problem. It has been defined as knowledge change and has been characterized for instance by the well-known AGM postulates [1] for the revision of belief sets, or by the Hansson's postulates [11,13] for the revision of belief bases. Several works have

E. Fermé and J. Leite (Eds.): JELIA 2014, LNAI 8761, pp. 442–456, 2014.

been proposed for the revision of *DL-Lite* knowledge bases (e.g. [21,9,12]), and especially for the revision of the ABox, since *DL-Lite* has witnessed a well suitability for Ontology-Based Data Access applications (OBDA). In such setting a TBox acts as being a schema used to reformulate raised queries in order to offer a better access to the set of data stored in an ABox.

Recently, an assertional-based "Removed Sets Revision" (RSR) approach has been proposed in [4] to revise *DL-Lite* knowledge bases. This approach is inspired from belief base revision in propositional logic framework [23,16]. It is based on inconsistency minimization, and consists in determining the smallest subsets of assertions which should be dropped from the current base in order to accept the new information and restore consistency. Note that in this approach, the minimality is understood with respect to cardinality and not with respect to set inclusion. The computation of the set of minimal assertions responsible of conflicts can be performed in polynomial time.

In real word applications, data is often provided by several and potentially conflicting sources. Their concatenation leads to a prioritized or a stratified ABox. This stratification generally results from two situations, as pointed out in [5]. 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 the sources, reflecting their reliability. The other situation 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.

To illustrate this situation, let us give the following example, adapted from [9]. Let K be a consistent knowledge base storing knowledge of an online newspaper collected using RSS feeds or Web crawling. The terminological base of this newspapers is as follows: wives are exactly those individuals who have husbands and some wives are employed. The assertional base A comes from crawling three distinct Web sources A_1, A_2 and A_3 where A_2 is more reliable than A_1 and A_1 is more reliable than A_3. A_1 says that Mary is a wife, A_2 says that Mary is employed and A_3 says that Mary's husband is John. It is clear that connecting information issued from A_1, A_2 and A_3 gives a prioritized assertional base. Assume that we found out an information to be incorporated into the knowledge base, which states that singles cannot be husbands. One can easily check that this new information not conflicting with the old ones stored in the knowledge base. Assume now that we found out another information saying that John is now single. One can verifies that this new information conflicts with the previous one. An important question addressed here is : "how one can we revise the knowledge base, while taking into account priorities between the assertions?".

The role of priorities in belief revision is very important and it has been largely studied in the literature, in the case where knowledge bases are encoded in a propositional logic setting (e.g. [6]). The notion of priorities in DLs is used in (e.g.[19]) to deal with defaults terminology while assuming that the ABox is completely sure. In [18] priorities are used to deal with inconsistencies in DL knowledge bases. In [17] the notion of priority has been used for ontology matching. Note that in [17] priorities are used on the set of concepts name and not on formulas. However, as far as we know, revising of prioritized assertional-based in *DL-Lite* knowledge bases has not been addressed so far.

This paper goes one step further in the definition of assertional-based RSR and investigates revision when priorities attached to assertions are available. This extension is based on the notion of Prioritized Removed Sets proposed in [3] for revising a set of prioritized propositional formulas. The minimality in revision with prioritized removed set refers to the lexicographic criterion and not to the set inclusion one. In this paper, we study revision for different forms of input: an ABox assertion or a TBox axiom. We define prioritized removed sets in *DL-Lite* framework. The main contribution of this work is to analyze the different scenarios of revision. In particular, we show that for some form of conflicts and some kinds of inputs, the revision process can be achieved in polynomial time. In the general case, we show that the number of prioritized removed sets is bounded and we propose an algorithm for computing them using the notion of hitting sets.

The rest of this paper is organized as follows. Section 2 gives brief preliminaries on *DL-Lite* logic. Section 3 investigates prioritized assertional-based removed sets revision within the framework of *DL-Lite* and gives the logical properties of the proposed operators. Section 4 provides algorithms for computing the prioritized removed sets through the use of hitting sets. We show that in particular cases revision process can be performed in a polynomial time. Section 5 concludes this paper.

2 Preliminaries

In this paper, we only consider $DL\text{-}Lite_R$, which underlies *OWL2-QL*. However, results of this work can be easily generalized for others DL-Lite logics (see [2] for more details about the *DL-Lite* family).

Syntax A *DL-Lite* knowledge base $\mathcal{K}=\langle\mathcal{T},\mathcal{A}\rangle$ is built upon a set of atomic concepts (i.e. unary predicates), a set of atomic roles (i.e. binary predicates) and a set of individuals (i.e. constants). Complex concepts and roles are formed as follows:

$$B \longrightarrow A|\exists R \quad C \longrightarrow B|\neg B$$
$$R \longrightarrow P|P^- \quad E \longrightarrow R|\neg R$$

where A (*resp.* P) is an atomic concept (*resp.* role). B (*resp.* C) are called basic (*resp.* complex) concepts and roles R (*resp.* E) are called basic (*resp.* complex) roles. The TBox $\mathcal{T}$ consists of a finite set of *inclusion axioms between concepts* of the form: $B \sqsubseteq C$ and *inclusion axioms between roles* of the form: $R \sqsubseteq E$. The ABox $\mathcal{A}$ consists of a finite set of *membership assertions* on atomic concepts and on atomic roles of the form: $A(a_i)$, $P(a_i, a_j)$, where a_i and a_j are individuals. For the sake of simplicity, in the rest of this paper, when there is no ambiguity we simply use *DL-Lite* instead of $DL\text{-}Lite_R$.

Semantics The *DL-Lite* semantics is given by an interpretation $I = (\Delta, .^I)$ which consists of a nonempty domain Δ and an interpretation function $.^I$. The function $.^I$ assigns to each individual a an element $a^I \in \Delta^I$, to each concept C a subset $C^I \subseteq \Delta^I$ and to each role R a binary relation $R^I \subseteq \Delta^I \times \Delta^I$ over Δ^I. Moreover, the interpretation function $.^I$ is extended for all constructs of $DL\text{-}Lite_R$. For instance: $(\neg B)^I=\Delta^I \backslash B^I$, $(\exists R)^I=\{x \in \Delta^I | \exists y \in \Delta^I \, such\, that\, (x, y) \in R^I\}$ and $(P^-)^I=\{(y, x) \in \Delta^I \times$

$\Delta^I|(x,y) \in P^I\}$. Concerning the TBox, we say that I satisfies a concept (*resp.* role) inclusion axiom, denoted by $I \models B \sqsubseteq C$ (*resp.* $I \models R \sqsubseteq E$), iff $B^I \subseteq C^I$ (*resp.* $R^I \subseteq E^I$). Concerning the ABox, we say that I satisfies a concept (*resp.* role) membership assertion, denoted by $I \models A(a_i)$ (*resp.* $I \models P(a_i, a_j)$), iff $a_i^I \in A^I$ (*resp.* $(a_i^I, a_j^I) \in P^I$). Note that we only consider *DL-Lite* with unique name assumption. Finally, an interpretation I is said to satisfy a knowledge base $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ iff I satisfies every axiom in $\mathcal{T}$ and every axiom in $\mathcal{A}$. Such interpretation is said to be a model of $\mathcal{K}$.

Incoherence and inconsistency Two kinds of inconsistency can be distinguished in DL-based knowledge bases: incoherence and inconsistency [4]. 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 iff it does not admit any model and it is said to be incoherent if there exists at least a non-satisfiable concept, namely for each interpretation I which is a model of $\mathcal{T}$, we have C^I=$\emptyset$.

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 by using the following rules repeatedly until reaching a fix point (see [8] 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 three such assertions are in $cln(\mathcal{T})$.

An important property has been established in [8] for consistency checking in *DL-Lite*. Formally, $\mathcal{K}$ is consistent if and only if $\langle cln(\mathcal{T}), \mathcal{A}\rangle$ is consistent [8].

3 PRSR for DL-Lite Knowledge Bases

In this section, we investigate *DL-Lite* prioritized knowledge base revision using a lexicographical strategy based on inconsistency minimization, well-known as Prioritized Removed Sets Revision (PRSR) [3], and previously defined in a classical logic setting.

3.1 Conflict Sets

Let $\mathcal{L}$ be a *DL-Lite* description language, presented in section 2 and $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$ be a *DL-Lite* prioritized knowledge base expressed in $\mathcal{L}$. We assume $\mathcal{T}$ is coherent and not stratified. On contrast, the ABox is stratified i.e. partitioned into n strata, $\mathcal{A}$=$\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n$ such that the assertions in $\mathcal{A}_i$ have the same level of priority and have higher priority than the ones in $\mathcal{A}_j$ where $j > i$. We assume that $\mathcal{K}$ is consistent and let us denote by N a new consistent 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 $DL\text{-}Lite_R$ language, N may be an assertions, a positive inclusion axiom (PI) or a negative inclusion axiom (NI). In some cases N may have a desirable interaction with $\mathcal{K}$. Clearly, according to [8], every *DL-Lite* knowledge base $\mathcal{K}$ with only PIs in its TBox is always satisfiable (consequence of Lemma 7 in [8]). However when the TBox $\mathcal{T}$ contains NI axioms then N may have an undesirable interaction with $\mathcal{K}$, which leads to inconsistency. In this case, a natural question for revising $\mathcal{K}$ is: which of the TBox axioms or ABox assertions should be removed first with respect to some ABox, since a TBox may be incoherent but never inconsistent. We remind the Calvanese *et al.* result [9].

Lemma 1. *Let $\mathcal{K} = \langle\mathcal{T},\mathcal{A}\rangle$ be a DL-Lite knowledge base. If $\mathcal{A} = \emptyset$ then $\mathcal{K}$ is consistent. If $\mathcal{K}$ is inconsistent, then there exists a subset $\mathcal{A}' \subseteq \mathcal{A}$ with at most two elements, such that $\mathcal{T} \cup \mathcal{A}'$ is inconsistent.*

In this paper, revision leads to ignoring some assertions, namely we give a priority to the TBox over ABox. Furthermore we only focus on inconsistency and assume that $\mathcal{T}$ is coherent and not stratified. This is not a restriction. This particular case can be handled outside the revision problem considered in this paper. Recall that this choice is motivated by the fact that *DL-Lite* framework was especially tailored for Ontology-Based based Access setting, in which the TBox is needed to access to the data stored in the ABox. Let $\mathcal{K}$ be an inconsistent knowledge base, we define the notion of conflict as a minimal inconsistent subset of $\mathcal{A}$, more formally:

Definition 1. *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: $i)$ $C \subseteq \mathcal{A}$, $ii)$ $\langle\mathcal{T},C\rangle$ is inconsistent, $iii)$ $\forall C'$, $C' \subset C$, $\mathcal{T} \cup C'$ 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.

Within the *DL-Lite* framework, in order to restore consistency while keeping new information, the Prioritized Removed Sets Revision strategy removes exactly one assertion in each conflict, by choosing the minimum number of assertions from $\mathcal{A}_1$, then the minimum number of assertions in $\mathcal{A}_2$, and so on. Using lexicographic criterion instead of set inclusion one reduces the set of potential conflicts. Taking the stratification of the $ABox$ into account has not been considered before for revising or repairing *DL-Lite* knowledge bases (*e.g.* [15,7]).

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

Definition 2. *Let X and X' be two subsets of $\mathcal{A} = \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$. X is strictly preferred to X', denoted by $X <_{lex} X'$ if and only if $i)$ $\exists i$, $1 \leq i \leq n$, $|X \cap \mathcal{A}_i| < |X' \cap \mathcal{A}_i|$, $ii)$ $\forall j$, $1 \leq j < i$, $|X \cap \mathcal{A}_j| = |X' \cap \mathcal{A}_j|$.*

Example 1. 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 $X <_{lex} X'$.

Definition 3. *let X and X' be two subsets of $\mathcal{A}$. X is at least equally preferred[1] to X', denoted by $X \leq_{lex} X'$ if and only if : i) $\exists i$, $1 \leq i \leq n$, $|X \cap \mathcal{A}_i| \leq |X' \cap \mathcal{A}_i|$, ii) $\forall j$, $1 \leq j < i$, $|X \cap \mathcal{A}_j| = |X' \cap \mathcal{A}_j|$.*

We now more formally present PRSR according to the nature of the input information.

3.2 Revision by a Membership Assertion

We first consider the case where N is a membership assertion. It corresponds to the revision by a fact or by an observation. In what follows, $\mathcal{K} \cup \{N\}$ denotes $\langle \mathcal{T}, \mathcal{A} \cup \{N\} \rangle$ where $\mathcal{A}$ is a prioritized ABox. The following definition introduces the concept of prioritized removed set.

Definition 4. *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 $i)$ $X \subseteq \mathcal{A}$, $ii)$ $\langle \mathcal{T}, (\mathcal{A} \backslash X) \cup \{N\} \rangle$ is consistent, $iii)$ $\forall X' \subseteq \mathcal{A}$, if $\langle \mathcal{T}, (\mathcal{A} \backslash X') \cup \{N\} \rangle$ is consistent then $X \leq_{lex} X'$.*

We denote by $\mathcal{PR}(\mathcal{K} \cup \{N\})$ the set of prioritized removed sets of $\mathcal{K} \cup \{N\}$. If $\mathcal{K} \cup \{N\}$ is consistent then $\mathcal{PR}(\mathcal{K} \cup \{N\}) = \emptyset$.

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 4, $X \subseteq \mathcal{A}$, $X' \subseteq \mathcal{A}$, $X =_{lex} X'$ and $\forall C \in \mathcal{C}(\mathcal{K} \cup \{N\})$, $C \cap X \neq \emptyset$ and $C \cap X' \neq \emptyset$. Moreover $C \cap \{N\} \neq \emptyset$, therefore $|C \cap \mathcal{A}| = 3$ which contradicts lemma 1. □

Definition 5. *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 defined by $\mathcal{K} \circ_{PRSR} N = \langle \mathcal{T}, \mathcal{A} \circ_{PRSR} N \rangle$ where $\mathcal{A} \circ_{PRSR} N = (\mathcal{A} \backslash X) \cup \{N\}$ with $X \in \mathcal{PR}(\mathcal{K} \cup \{N\})$.*

When N is a membership assertion and the ABox is prioritized, PRSR gives the same result as RSR[4] in the flat case (where all the assertions in the ABox have the same priority). More formally:

Proposition 2. *Let $\mathcal{K}$ be a consistent stratified knowledge base and N be a membership assertion. $\mathcal{K} \circ_{PRSR} N = \mathcal{K} \circ_{RSR} N$.*

[1] X is equally preferred to X', denoted by $X =_{lex} X'$, iff $X \leq_{lex} X'$ and $X' \leq_{lex} X$.

Proof (Sketch of proof). The proof is immediate. It follows from Proposition 1, since $|\mathcal{PR}(\mathcal{K} \cup \{N\})| = 1$. □

Example 2. 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 1, $\mathcal{C}(\mathcal{K} \cup \{N\}) = \{\{B_1(a), B_3(a)\}, \{B_3(a), B_4(a)\}\}$. Hence, by Definition 4, $\mathcal{PR}(\mathcal{K} \cup \{N\}) = \{\{B_1(a), B_4(a)\}\}$. Therefore $\mathcal{A} \circ_{PRSR} N = \{B_3(b), B_3(a)\}$.

As detailed in [4], computing the set of conflicts is polynomial. Moreover when the input is a membership assertion, as illustrated in the above example, Proposition 1 states that there is only one prioritized removed set, which is computed in polynomial time as shown in Section 4.

3.3 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. In this case, $\mathcal{K} \cup \{N\}$ denotes $\langle\mathcal{T} \cup \{N\}, \mathcal{A}\rangle$.

Definition 6. *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 $i)$ $X \subseteq \mathcal{A}$, $ii)$ $\langle\mathcal{T} \cup \{N\}, (\mathcal{A} \backslash X)\rangle$ is consistent and $iii)$ $\forall X' \subseteq \mathcal{A}$, if $\langle\mathcal{T} \cup \{N\}, (\mathcal{A} \backslash X')\rangle$ is consistent then $X \leq_{lex} X'$.*

Let us point out that Definition 6 is similar to Definition 4, except that new information is not added to the ABox but to the TBox. However, the revision process still considers the TBox as a stable knowledge. Therefore, in order to restore consistency, assertional elements 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 3. Let $\mathcal{K}$=$\langle\mathcal{T}, \mathcal{A}\rangle$ be a consistent stratified knowledge base such that $\mathcal{T}$=$\{B_1 \sqsubseteq B_2, B_3 \sqsubseteq \neg B_4\}$ 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 removed sets [4] are $X_1 = \{B_1(a), B_2(b)\}$, $X_2 = \{B_1(a), B_3(b)\}$, $X_3 = \{B_3(a), B_2(b)\}$, $X_4 = \{B_3(a), B_3(b)\}$ however there is only one prioritized removed set X_4 as illustrated in table 1.

Table 1. One prioritized removed set

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

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

Table 2. Two prioritized removed sets

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

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 following proposition provides the condition of the existence of exactly one prioritized removed set.

Proposition 3. *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 that there are two prioritized removed sets, X and X', with $X \neq X'$. By Definition 6 $X \subseteq \mathcal{A}$, $X' \subseteq \mathcal{A}$, $X =_{lex} X'$ and $\forall C \in \mathcal{C}(\mathcal{K} \cup \{N\})$, $C \cap X \neq \emptyset$ 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. If $|C \cap X| = 1$ and $|C \cap X'| = 1$ then 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 $C \cap \mathcal{A}_j \neq \emptyset$ which contradicts $X =_{lex} X'$. If $C \cap X = C \cap X'$ then $X = X'$ which contradicts the hypothesis. □

This situation holds when each stratum is consistent with $\mathcal{T} \cup \{N\}$, for example when the stratification comes from several experts with different degrees of reliability. In this case, as detailed in section 4, computing the unique prioritized removed set is polynomial. The following proposition gives the condition of existence of several prioritized removed sets.

Proposition 4. *If there exists $C \in \mathcal{C}(\mathcal{K} \cup \{N\})$ such that there exists i, $C \cap \mathcal{A}_i \neq \emptyset$ and for all j, $j \neq i$, $C \cap \mathcal{A}_j = \emptyset$, then $|\mathcal{PR}(\mathcal{K} \cup \{N\})| \geq 2$.*

Proof. Suppose there is only one prioritized removed set X. By Definition 6, $X \subseteq \mathcal{A}$ and $C \cap X \neq \emptyset$. If $|C \cap X| = 2$ then X is not a prioritized removed set. If $|C \cap X|$=1, since there exists i, such that $|C \cap \mathcal{A}_i|$=2 therefore there exists X and X' such that $C \cap X \neq \emptyset$ and $C \cap X' \neq \emptyset$ and $X =_{lex} X'$ which contradicts the hypothesis. □

There are several prioritized removed sets as soon as there are conflicts included in a stratum where each conflict may leads 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 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 $X_i \in \mathcal{PR}(\mathcal{K} \cup \{N\})$. In the *DL-Lite* language, it is not possible to find a knowledge base which represents the disjunction of such possible revised knowledge base. If we want to keep the result of revision in *DL-Lite*, several options are possible. The first one is to consider the intersection of all possible revised knowledge bases however this option may be too cautious since it could remove too many assertions and contradicts in some sense the minimal change principle. Another option is to define a selection function, where the revised knowledge base is defined as follows.

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. 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 $\mathcal{A}\circ_{PRSR} N$=$(\mathcal{A}\backslash f(\mathcal{PR}(\mathcal{K}\cup\{N\})))$.*

When N is a NI or a PI axiom, PRSR generalizes RSR [4]. More formally:

Proposition 5. *Let $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$ be a consistent knowledge base, N be a PI or a NI axiom. If $\mathcal{A}$ is not stratified then $\mathcal{K}\circ_{PRSR} N = \mathcal{K}\circ_{RSR} N$*

Proof. If $\mathcal{A}$ is not stratified, i.e. there is only one stratum, conditions $i)$ and $ii)$ in Definition 6 do not change and condition $iii)$ becomes $\forall\subseteq\mathcal{A}$, if $\langle\mathcal{T}\cup\{N\},(\mathcal{A}\backslash X')\rangle$ is consistent, then $|X\cap\mathcal{A}| < |X'\cap\mathcal{A}|$ since $X\subseteq\mathcal{A}$. It follows that $\forall\subseteq\mathcal{A}$ if $\langle\mathcal{T}\cup\{N\},(\mathcal{A}\backslash X')\rangle$ is consistent then $|X| < |X'|$, which is the third condition in the definition of a removed set [4]. □

3.4 Logical Properties

Revision within the framework of Description logics, in particular *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 [11,13]. In [4] the Hansson's postulates are rephrased within DL-Lite framework.

Let $\mathcal{K}$, $\mathcal{K}'$ be *DL-Lite* knowledge bases, N and M be either membership assertions or positive or negative axioms, $\circ$ be a revision operator. $\mathcal{K}+N$ denotes the non closing expansion, i.e. $\mathcal{K}+N$=$\mathcal{K}\cup\{N\}$. The postulates are: **P1 (Success)** $N\in\mathcal{K}\circ N$. **P2 (Inclusion)** $\mathcal{K}\circ N\subseteq\mathcal{K}+N$. **P3 (Consistency)** $\mathcal{K}\circ N$ is consistent. **P4 (Vacuity)** If $\mathcal{K}\cup\{N\}$ is consistent then $\mathcal{K}\circ N$=$\mathcal{K}+\{N\}$. **P5 (Pre-expansion)** $(\mathcal{K}+N)\circ N$=$\mathcal{K}\circ N$. **P6 (Internal exchange)** If N, $M\in\mathcal{K}$ then $\mathcal{K}\circ N$=$\mathcal{K}\circ M$. **P7 (Core retainment)** If $M\in\mathcal{K}$ and $M\notin\mathcal{K}\circ N$ then there is at least one $\mathcal{K}'$ such that $\mathcal{K}'\subseteq\mathcal{K}+N$, and $\mathcal{K}'$ is consistent but $\mathcal{K}'\cup\{M\}$ is inconsistent. **P8 (Relevance)** If $M\in\mathcal{K}$ and $M\notin\mathcal{K}\circ N$ then there is at least one $\mathcal{K}'$ such that $\mathcal{K}\circ N\subseteq\mathcal{K}'\subseteq\mathcal{K}+N$, and $\mathcal{K}'$ is consistent but $K'\cup\{M\}$ is inconsistent.

Proposition 6. *Let $\mathcal{K}$ be a consistent stratified knowledge base. If N is a membership assertion then the revision operator $\circ_{PRSR}$ satisfies the postulates* **P1**- **P8**. *If N is a PI or a NI axiom then the revision operator $\circ_{PRSR}$ satisfies the postulates* **P1**- **P7**.

Proof (Sketch of proof). For both revision operators **P1**-**P6** follow from the definition of PRSR and **P7** follows from the existence of at least one prioritized removed set. On contrast **P8** requires the existence of only one prioritized removed set, which is the case when N is a membership assertion, but this is not the case in general when N is a PI or a NI axiom, except for the case stated in Proposition 3. □

In the next section, we provide different algorithms for computing the prioritized removed sets depending on the nature of the input.

4 Computing Revision Operation

As stated before, when trying to revise a *DL-Lite* knowledge base 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, the content of the knowledge base and the form of the conflicts.

4.1 Result of Revision by an Assertion

When new information is an assertion, thanks to Proposition 1, there exists only one prioritized removed set. The computation of this set amounts in picking in each conflict the assertion which is different from the input N. This operation follows from a simple and non costly adaptation of the algorithm given in [8] for checking the consistency of a *DL-Lite* knowledge base. The main difference is that in [8] the aim is only to check whether a *DL-Lite* knowledge base is consistent or not. Here, we do one step further, as we need to enumerate all assertional facts that conflict with the input. Computing these conflicting assertions with N first requires the negative closure $cln(\mathcal{T})$, computed using the rules given in Section 2 repetively until reaching a fixed point. We suppose that this is performed by a NEGCLOSURE function. We provide the algorithm COMPUTEPRSR1, which computes the prioritized removed set $PR \in \mathcal{PR}(\mathcal{K} \cup \{N\})$.

Algorithm 1. COMPUTEPRSR1

function COMPUTEPRSR1 ($\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle, N$)
    $PR \leftarrow \emptyset$
    $cln(\mathcal{T}) \leftarrow$ 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**
                $PR \leftarrow PR \cup \{\alpha\}$
    Return PR

Generally, the computation of the conflicts proceeds with the evaluation over $\mathcal{A}$ of each NI axiom in $cln(\mathcal{T})$ in order to exhibit whether $\mathcal{A}$ contains assertions which contradict the NI axioms. Intuitively, for each $X \sqsubseteq \neg Y$ belonging to $cln(\mathcal{T})$, the evaluation of $X \sqsubseteq \neg Y$ over the $\mathcal{A}$ simply amounts to return all $(X(x), Y(x))$ such that $X(x)$ and $Y(x)$ belongs to $\mathcal{A}$. When N is an assertion, one can easily check that every conflict which 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 removed set computation can be performed in polynomial time.

Note that the algorithm COMPUTEPRSR1 produces the same revision result as the algorithm proposed in [9], since revision with an ABox assertion is uniquely defined (theorem 13 in [9]).

4.2 PRSR Computation : Revision by an Axiom

We now detail the case where N is a PI or a NI axiom. According to Definition 6, computing $\mathcal{PR}(\mathcal{K} \cup \{N\})$ starts with the computation of $\mathcal{PR}((\mathcal{T} \cup \{N\}) \cup \mathcal{A}_1)$, then continues with the computation $\mathcal{PR}((\mathcal{T} \cup \{N\}) \cup (\mathcal{A}_1 \cup \mathcal{A}_2))$, and so on. A prioritized removed set is formed by picking in each conflict the least priority element. However, according to the form of conflicts, two situations hold as pointed out in Section 3.

The first one is when each conflict involves two elements having different levels of priority. From Proposition 3, we have shown that there exists only one prioritized removed set. We provide the algorithm COMPUTEPRSR2 which computes the prioritized removed set $PR \in \mathcal{PR}(\mathcal{K} \cup \{N\})$.

Algorithm 2. COMPUTEPRSR2

function COMPUTEPRSR2 ($\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle, N$)
    $\mathcal{T}' \leftarrow \mathcal{T} \cup \{N\}, \mathcal{K}' = \langle \mathcal{T}', \mathcal{A} \rangle$
    $cln(\mathcal{T}') \leftarrow$ NEGCLOSURE($\mathcal{T}'$)
    $PR \leftarrow \emptyset$
    $i \leftarrow 1$
    while $i \leq n$ **do**
        for all $X \sqsubseteq \neg Y \in cln(\mathcal{T}')$ **do**
            for all $\alpha \in \mathcal{A}_i$ **do**
                $j \leftarrow i + 1$
                while $j \leq n$ **do**
                    for all $\beta \in \mathcal{A}_j$ **do**
                        if $\langle X \sqsubseteq \neg Y, \{\alpha, \beta\} \rangle$ is inconsistent **then**
                            $PR \leftarrow PR \cup \{\beta\}$
                            $\mathcal{A}_j \leftarrow \mathcal{A}_j \setminus \{\beta\}$
                    $j \leftarrow j + 1$
        $i \leftarrow i + 1$
    Return PR

The algorithm COMPUTEPRSR2 proceeds from a current layer to all the other less preferred layers and selects the assertions which conflict with the ones in the current layer. We increment from a layer to another in order to ensure the minimality of the prioritized removed set w.r.t. lexicographic ordering. Note that this algorithm is based on inconsistency checking and its computational complexity is polynomial.

We now describe the second case, where there exists at least a conflict involving two elements having the same priority level. In such situation there are several prioritized removed sets to be computed, as pointed out in Proposition 4. In order to compute them, we follow the idea proposed in [23], where removed sets in the flat case can be computed using the hitting set notion [20]. A hitting set is a set which intersects each set in a collection. A minimal hitting set, w.r.t. set inclusion, is called a kernel. Moreover, kernels which are minimal w.r.t. cardinality correspond to the definition of a removed set [23]. The same result has been established for the removed set revision of *DL-Lite* knowledge bases [4] where the computation of the kernels of $\mathcal{C}(\mathcal{K} \cup \{N\})$ is performed using Reiter's algorithm [20], modified in [22]. We recall this algorithm [4].

Definition 8. *A tree T is an HS-tree of $\mathcal{C}(\mathcal{K} \cup \{N\})$ if and only if it is the smallest tree having the following properties:*

1. *Its root is labeled by an element from $\mathcal{C}(\mathcal{K} \cup \{N\})$. If $\mathcal{C}(\mathcal{K} \cup \{N\})$ is empty, its root labeled by '$\surd$'.*
2. *If m is a node from T, let $H(m)$ be the set of branch labels on the path going from the root to T to m. If m is labeled by '$\surd$', it has no successor in T.*
3. *If m is labeled by a set $C \in \mathcal{C}(\mathcal{K} \cup \{N\})$, then, for each $c \in C$, m has a successor node m_c in T, joined to m by a branch labeled by c. The label of m_c is a set $C^{'} \in \mathcal{C}(\mathcal{K} \cup \{N\})$ such that $C^{'} \cap H(m_c) = \emptyset$, if such a set exist. Otherwise, m_c is labeled by '$\surd$'.*

The kernels correspond to the leaves labeled by $\surd$. For each such node m, $H(m)$ is a kernel of $\mathcal{C}(\mathcal{K} \cup \{N\})$. We use the same pruning techniques as in [22].

Concerning prioritized removed sets, they are not necessarily minimal w.r.t. cardinality. But they are minimal w.r.t. lexicographic ordering ($\leq_{lex}$ for short). So, a naive algorithm for computing $\mathcal{PR}(\mathcal{K} \cup \{N\})$ is : (i) compute the kernels of $\mathcal{C}(\mathcal{K} \cup \{N\})$. (ii) keep only minimal ones w.r.t. $\leq_{lex}$. However, we can improve the algorithm.

As we said before, a prioritized removed set is computed from one layer to another. The idea of the enhancement of the algorithm is as follows: First, compute the conflicts in the first layer, i.e. in $(\mathcal{T} \cup \{N\}) \cup A_1$, then build the hitting set tree on this collection of conflicts. This tree allows for the computation of the kernels of $(\mathcal{T} \cup \{N\}) \cup A_1$ minimal w.r.t. $\leq_{lex}$. From these kernels, continue the construction of the tree using conflicts in $(\mathcal{T} \cup \{N\}) \cup (\{A_1 \cup A_2\})$ if they exist, and so on until reaching a fixed point where no conflict will be generated. Now the kernels of the final hitting set tree using conflicts in $(\mathcal{T} \cup \{N\}) \cup (\{A_1 \cup A_2 \cup ... \cup A_n\})$ which are minimal w.r.t. $\leq_{lex}$ are the prioritized removed sets. The following algorithm COMPUTEPRSR3 computes $\mathcal{PR}(\mathcal{K} \cup \{N\})$ using hitting sets.

Algorithm 3. COMPUTEPRSR3

function COMPUTEPRSR3 ($\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$, N)
    $\mathcal{T}' \leftarrow \mathcal{T} \cup \{N\}, \mathcal{K}' = \langle \mathcal{T}', \mathcal{A} \rangle$
    $cln\,(\mathcal{T}') \leftarrow$ NEGCLOSURE($\mathcal{T}'$)
    $\mathcal{PR}(\mathcal{K}') \leftarrow \emptyset$
    $\mathcal{C} \leftarrow \emptyset$, TREE$\leftarrow \emptyset$, $i \leftarrow 1$
    while $i \leq n$ **do**
        for all $X \sqsubseteq \neg Y \in cln\,(\mathcal{T}')$ **do**
            for all (α, β) s.t. $\alpha \in \mathcal{A}_1, \beta \in \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_i$ **do**
                if $\langle X \sqsubseteq \neg Y, \{\alpha, \beta\} \rangle$ is inconsistent **then**
                    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\alpha, \beta\}$
        TREE$\leftarrow$ TREE.ADDFROMLEXKERNEL(HS($\mathcal{C}$))
        $\mathcal{C} \leftarrow \emptyset$,
        $i \leftarrow i + 1$
    $\mathcal{PR}(\mathcal{K}') \leftarrow$ LEXKERNEL(TREE)
    Return $\mathcal{PR}(\mathcal{K}')$

In this algorithm the function HS($\mathcal{C}$) takes as input the conflicts computed in each layer (if they exist) and builds the corresponding hitting sets tree (TREE), using the algorithm presented in Definition 8. From a layer to another, we resume the construction of (TREE) from its current kernels minimal w.r.t. $\leq_{lex}$. Namely, the function ADDFROMLEXKERNEL((HS($\mathcal{C}$)) builds the hitting set tree of a collection of conflicts $\mathcal{C}$ starting from the kernels branches of the current TREE which are minimal w.r.t. $\leq_{lex}$. Finally $\mathcal{PR}(\mathcal{K} \cup \{N\})$ corresponds to the kernels of TREE obtained using function LEXKERNEL(TREE)) which are minimal w.r.t. $\leq_{lex}$. Note that COMPUTEPRSR3 is a generalization of COMPUTEPRSR2, since when all conflicts involve elements from distinct layers, then the final tree will only contains one prioritized removed set. The following example illustrates this algorithm.

Example 4. 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 conflicts obtained from $cln(\mathcal{T}') \cup \mathcal{A}_1$ are $\{A(a), D(a)\}$. The constructed tree using HS($\{A(a), D(a)\}$) will contain two branches labeled respectively by $A(a)$ and $D(a)$ which are kernels minimal w.r.t. $\leq_{lex}$ ($\leq_{lex}$ kernel). We continue with $cln(\mathcal{T}') \cup \mathcal{A}_1 \cup \mathcal{A}_2$ where $\{C(a), D(a)\}$ is a conflict. We resume the construction of the tree its current $\leq_{lex}$ kernel (branches labeled by $A(a)$ and $D(a)$) and we obtain three HS-tree: $\{A(a), C(a)\}$, $\{A(a), D(a)\}$ and $D(a)$ where only $D(a)$ is $\leq_{lex}$ kernel. Now, we increment to $cln(\mathcal{T}') \cup \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ where $\{B(b), D(b)\}$ is a conflict and we continue the construction of the Tree from $D(a)$. We obtain $\{D(a), D(b)\}$ and $\{D(a), B(b)\}$ as HS-tree where only $\{D(a), D(b)\}$ is $\leq_{lex}$ kernel. Finally, We we have $\{D(c), C(c)\}$ as a conflict in $cln(\mathcal{T}') \cup \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4$. We continue the construction of the tree from 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, $\mathcal{PR}(\mathcal{K} \cup \{N\})$=$\{D(a), D(b), C(c)\}$, $\{D(a), D(b), D(c)\}$.

5 Conclusion

In this paper, we investigated Prioritized Removed Sets Revision of *DL-Lite* knowledge bases. We studied the revision operation for three forms of input, namely, an ABox assertion or a TBox axiom. We first defined the prioritized removed sets within the framework of *DL-Lite* as a lexicographic approach. We showed that when the input is an assertion then PRSR is computed in polynomial time. When the input is a PI or a NI axiom we provided the condition for the computation of PRSR in polynomial time. We showed that in the general case the number of prioritized removed sets is bounded and we proposed an algorithm for computing these sets using the notion of hitting sets. We finally gave logical properties of the proposed operators in terms of satisfaction of Hansson's postulates rephrased in our framework. In a near future we plan to investigate the iterated revision of *DL-Lite* knowledge bases. We also want focus on the extension of Removed Sets Fusion [14], defined in a propositional setting, to the merging of *DL-Lite* knowledge bases.

Acknowledgement. This work has received support from the french Agence Nationale de la Recherche, ASPIQ project reference ANR-12-BS02-0003.

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 (JAIR) 36, 1–69 (2009)
3. 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)
4. Benferhat, S., Bouraoui, Z., Papini, O., Würbel, E.: Assertional-based removed sets revision of DL-Lite$_R$ knowledge bases. In: ISAIM (2014)
5. Benferhat, S., Dubois, D., Prade, H.: How to infer from inconsisent beliefs without revising. In: IJCAI, pp. 1449–1457 (1995)
6. Benferhat, S., Dubois, D., Prade, H., Williams, M.-A.: A practical approach to revising prioritized knowledge bases. Studia Logica 70(1), 105–130 (2002)
7. Bienvenu, M., Rosati, R.: New inconsistency-tolerant semantics for robust ontology-based data access. In: Eiter, T., Glimm, B., Kazakov, Y., Krötzsch, M. (eds.) Description Logics. CEUR Workshop Proceedings, vol. 1014, pp. 53–64. CEUR-WS.org (2013)
8. 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. Reasoning 39(3), 385–429 (2007)
9. 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.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 112–128. Springer, Heidelberg (2010)
10. 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.) ISWC 2005. LNCS, vol. 3729, pp. 216–231. Springer, Heidelberg (2005)
11. Fuhrmann, A.: An essay on contraction. CSLI Publications, Stanford (1997)
12. Gao, S., Qi, G., Wang, H.: A new operator for abox revision in dl-lite. In: Hoffmann, J., Selman, B. (eds.) AAAI. AAAI Press (2012)
13. Hansson, S.O.: Revision of belief sets and belief bases. Handbook of Defeasible Reasoning and Uncertainty Management Systems 3, 17–75 (1998)
14. Hué, J., Würbel, E., Papini, O.: Removed sets fusion: Performing off the shelf. In: Proc. of ECAI 2008 (FIAI 178), pp. 94–98 (2008)
15. Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., Savo, D.F.: Inconsistency-tolerant semantics for description logics. In: Hitzler, P., Lukasiewicz, T. (eds.) RR 2010. LNCS, vol. 6333, pp. 103–117. Springer, Heidelberg (2010)
16. Papini, O.: A complete revision function in propositional calculus. In: ECAI, pp. 339–343 (1992)
17. Qi, G., Ji, Q., Haase, P.: A conflict-based operator for mapping revision. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 521–536. Springer, Heidelberg (2009)
18. 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)
19. Qi, G., Pan, J.: A stratfication-based approach for inconsistency handling in description logics. In: IWOD 2007, p. 83 (2007)

20. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. 32(1), 57–95 (1987)
21. Wang, Z., Wang, K., Topor, R.W.: A new approach to knowledge base revision in dl-lite. In: Fox, M., Poole, D. (eds.) AAAI. AAAI Press (2010)
22. Wilkerson, R.W., Greiner, R., Smith, B.A.: A correction to the algorithm in reiter's theory of diagnosis. Artificial Intelligence 41, 79–88 (1989)
23. Würbel, E., Jeansoulin, R., Papini, O.: Revision: An application in the framework of GIS. In: Cohn, A.G., Giunchiglia, F., Selman, B. (eds.) KR, pp. 505–515. Morgan Kaufmann (2000)