# Assertional-based Removed Sets Revision of $DL\text{-}Lite_R$ Belief Bases

**Salem Benferhat** and **Zied Bouraoui**
Université d'Artois - Nord de France,
CRIL-CNRS UMR 8188,
{benferhat,bouraoui}@cril.univ-artois.fr

**Odile Papini**
Université d'Aix Marseille
LSIS-CNRS UMR 7296
odile.papini@univ-amu.fr

**Eric Würbel**
Université du Toulon
LSIS-CNRS UMR 7296
wurbel@univ-tln.fr

### Abstract

*DL-Lite* is a tractable fragment of Description Logics, which provides a powerful framework to compactly encode available knowledge with a low computational complexity of the reasoning process. Belief revision is an important problem that appears in many applications and especially in the semantic web area. This paper investigates the so-called "Removed Sets Revision" for revising *DL-Lite* knowledge bases when a new sure piece of information, called an input, is added. Such strategy of revision is based on inconsistency minimization and consists in determining smallest subsets of axioms that should be dropped from the current *DL-Lite* knowledge base in order to restore its consistency and accept the input. We consider different forms of the input: a positive inclusion axiom, a negative inclusion axiom and a membership assertion. We then show how to use the notion of hitting sets for computing removed sets. We show in particular, for some revision strategies and some kinds of inputs, that the removed sets revision can be achieved in polynomial time. Finally, we rephrase the Hansson's postulates for belief bases revision within a DL-Lite framework in order to give logical properties of removed sets revision operators.

## Introduction

Description Logics (DLs, for short) are well-known logics introduced for representing and reasoning about structured knowledge (Baader et al. 2003). In recent years, a lot of attention was given to *DL-Lite*, a family of tractable DLs investigated by (Calvanese et al. 2005). *DL-Lite* provides a powerful framework that allows a flexible representation of knowledge with a low computational complexity of the reasoning process. In particular, knowledge base consistency and all DLs standard reasoning services are polynomial for combined complexity (*i.e.* the overall size of the knowledge base) (Artale et al. 2009).

*DL-Lite* has an important place in various application areas such as the Ontology-Based Data Access and the Semantic Web where *DL-Lite* provides the logical underpinning of the *OWL2-QL* [1] language especially dedicated to applications that use huge volumes of data.

Originally the DLs have been introduced to represent the knowledge of a domain of interest in a static aspect (Baader et al. 2003). However in some web applications knowledge may not be static and evolves from a situation to another in order to adapt the different changes that occur over time.

Such dynamic aspects have been recognized as an important problem (Qi, Liu, and Bell 2006; Calvanese et al. 2010; Kharlamov and Zheleznyakov 2011; Bienvenu and Rosati 2013; Kharlamov, Zheleznyakov, and Calvanese 2013) and often concerns the situation where a 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 knowledge change and was characterized for instance by the well-known AGM postulates (Alchourrón, Gärdenfors, and Makinson 1985). These postulates focus on the logical structure of knowledge and are based on three main ideas:

- the principle of priority which states that the priority between beliefs is given to the new pieces of information,
- the consistency principle which states that the result of the revision operation must be a consistent set of beliefs, and
- the principle of minimal change which states that as least as possible beliefs (initial beliefs) should be changed in the revision operation.

Note that AGM postulates were defined for revising belief sets, i.e deductively closed sets of formulas, possibly infinite. However an axiomatic characterization for revision belief bases, i.e finite set of formulas was given in (Fuhrmann 1997; Hansson 1998).

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 (RSR), also known as a lexicographic-based approach, has been proposed in (Papini 1992; Benferhat et al. 1993) 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 criterion. This approach has interesting properties: it has not a high computational complexity, it is not too cautious and satisfies all rational AGM postulates for belief revision. An extension of RSR called Prioritized Removed Sets Revision (PRSR) (Benferhat et al. 2010) has been proposed for revising a set of prioritized propositional formulas.

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

Recently, several works have been proposed for revising DLs knowledge bases. In (Flouris, Plexousakis, and Antoniou 2004; 2005) an adaptation of the AGM theory in order to be generalized on DLs was discussed. In (Halaschek-wiener, Katz, and Parsia 2006; Ribeiro and Wassermann 2007) an extension of kernel-based revision and semi-revision operators to DLs frameworks has been proposed which are closely related to the one proposed by (Hansson 1997) in propositional logic setting.

In (Reiter 1987) the concept of "debugging" terminological bases has been introduced. The proposed solutions mainly adapt what has been proposed in diagnosis to general terminological knowledge bases. Regarding *DL-Lite* knowledge bases, few works have been proposed for the revision problem. In (Qi and Yang 2008; Qi and Du 2009; Wang, Wang, and Topor 2010), model-based approaches for revising DLs have been proposed. In (Calvanese et al. 2010; Kharlamov and Zheleznyakov 2011) a computational complexity analysis has been given for revising DL knowledge bases. In (Calvanese et al. 2010), a formula-based approach for *DL-Lite* knowledge bases revision has been presented. Two algorithms have been proposed one for revising TBox and the other for revising ABox. Another operator for ABox revision in *DL-Lite* based on graph structure has been introduced in (Gao, Qi, and Wang 2012). In this work, the new information is restricted to a membership assertion (ABox axiom).

This paper proposes an extension of removed sets revision, defined in a propositional setting to the case where knowledge bases are described in *DL-Lite* framework. One of the motivations in considering removed sets revision of *DL-Lite* knowledge bases is to take advantage on one hand of tractability of *DL-Lite* for the revision process and on the other hand of rational properties satisfied by the removed sets revision which has not been considered before. In particular we investigate the well-known $DL\text{-}Lite_R$ logic which offers a good compromise between expressivity power and computational complexity.

An important property when dealing with *DL-Lite* knowledge bases is that minimal membership assertional sets (minimality in the sense of cardinality and not in the sense of set inclusion) to be removed are either singletons or doubletons. This will be helpful in defining removed sets needed to restore consistency of a revised knowledge base. Another important feature when dealing with *DL-Lite* knowledge base is that computing the set of minimal information responsible of conflicts can be done in polynomial time.

The rest of this paper is organized as follows. Section 2 gives brief preliminaries on *DL-Lite* logic. Section 3 presents the extension of removed sets revision within the framework of *DL-Lite* knowledge bases. Section 4 provides algorithms for computing the removed sets through the use of hitting sets. Sections 5 reformulates the well-known Hansson's postulates within DL-Lite setting in order to study logical properties of removed sets revision operators. Section 6 discusses related works.

# A refresher on DL-Lite logic

For the sake of simplicity, we only consider $DL\text{-}Lite^{H}_{core}$ (known as $DL\text{-}Lite_R$) that underlies *OWL2-QL* language as *DL-Lite* logic. However results of this work can be easily adapted for others DL-Lite logics. For more details about different logics in *DL-Lite* family see (Artale et al. 2009).

## Syntax

The language of $DL\text{-}Lite_{core}$ is the core language for $DL\text{-}Lite_R$ (Calvanese et al. 2007) and it is formulated by a description language that forms concepts and roles 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 a TBox (Terminological Box) and $\mathcal{A}$ is an ABox (Assertional Box). The $DL\text{-}Lite_{core}$ TBox consists of a finite set of *inclusion assertions* of the form: $B \sqsubseteq C$.

The $DL\text{-}Lite_{core}$ ABox consists of a finite set of *membership assertions* on atomic concept and on atomic role of the form: $A(a_i)$ , $P(a_i, a_j)$, where $a_i$ and $a_j$ are two individuals. Note that we only consider *DL-Lite* with unique name assumption (*i.e.* $a_i \neq a_j$ where $i \neq j$).

The $DL\text{-}Lite_R$ extends $DL\text{-}Lite_{core}$ with the ability of specifying *inclusion assertion between roles* to the TBox of the form: $R \sqsubseteq E$.

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

As usual in DLs, the *DL-Lite* semantics is given by an interpretation $I = \left(\Delta, .^I\right)$ which consists of a nonempty domain $\Delta$ 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 $\Delta^I$. Moreover, the interpretation function $.^I$ is extended for all constructs of the $DL\text{-}Lite_R$. Namely: $(\neg B)^I = \Delta^I \backslash B^I$, $(\exists R)^I = \left\{x \in \Delta^I | \exists y \in \Delta^I \; such\, that \; (x, y) \in R^I\right\}$ and $(P^-)^I = \left\{(y, x) \in \Delta^I \times \Delta^I | \, (x, y) \in P^I\right\}$.

For the TBox, we say that $I$ satisfies a concept (*resp.* role) inclusion assertion, denoted by $I \models B \sqsubseteq C$ (*resp.* $I \models R \sqsubseteq E$), if and only if $B^I \subseteq C^I$ (*resp.* $R^I \subseteq E^I$).

For the ABox, we say that $I$ satisfies a concept (*resp.* role) membership assertion, denoted by $I \models A(a_i)$ (*resp.* $I \models P(a_i, a_j)$ ), if and only if $a_i^I \in A^I$ (*resp.* $\left(a_i^I, a_j^I\right) \in P^I$ ).

Lastly, an interpretation $I$ is said to satisfy a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ if and only if $I$ satisfies every axiom in $\mathcal{T}$ and every axiom in $\mathcal{A}$. Such interpretation is said to be a model of $\mathcal{K}$ (Calvanese et al. 2007).

## Incoherence and inconsistency

Two kinds of inconsistency can be distinguished in DL-based knowledge bases: incoherence and inconsistency (Baader et al. 2003; Flouris et al. 2006). 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 a model and it is said to be incoherent if there exists at least a non-satisfiable concept. More formally:

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

An 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 $I$ of $\mathcal{T}$ we have $B_1^I = \emptyset$. In a propositional setting the counterpart of incoherence is a so-called potential inconsistency defined for instance in (Nonfjall and Larsen 1992).

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 a model.*

In *DL-Lite* a TBox $\mathcal{T} = \{PIs, NIs\}$ can be viewed as composed of positive inclusion assertions, denoted by (PIs), and negative inclusion assertions, denotes 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 $R_1 \sqsubseteq \neg R_2$. Conceptually, the *PIs* (*resp.* NIs) represent subsumption (*resp.* disjunction) between concepts or roles. 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. The $cln(\mathcal{T})$ is obtained using the following rules (see (Calvanese et al. 2007) for more details) repetitively until reaching a fix point:

- 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 $cln(\mathcal{T})$.

In *DL-Lite*, a new property was given for consistency checking. Formally, $\mathcal{K}$ is said to be consistent if and only if $\langle cln(\mathcal{T}), \mathcal{A} \rangle$ is consistent (Calvanese et al. 2007).

## RSR for DL-Lite knowledge bases

In this section, we investigate the problem of *DL-Lite* knowledge base revision using a strategy based on inconsistency minimization well-known as Removed Sets Revision (RSR) (Papini 1992).

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* knowledge base expressed in $\mathcal{L}$. We assume that $\mathcal{K}$ is consistent and $\mathcal{T}$ is coherent. 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 a membership assertion of the form $A(a)$ or $P(a, b)$, a positive inclusion assertion (PI) of the form $B_1 \sqsubseteq B_2$ or a negative inclusion assertion (NI) of the form $B_1 \sqsubseteq \neg B_2$.

In some cases $N$ may have a desirable interaction with $\mathcal{K}$. Clearly, according to (Calvanese et al. 2007), every *DL-Lite* knowledge base $\mathcal{K}$ with only PIs in its TBox is always satisfiable (consequence of Lemma 7 in (Calvanese et al. 2007)). Hence, if $N$ is a membership assertion or a PI assertion, there is no inconsistency. However when the TBox $\mathcal{T}$ contains NI assertions then $N$ may have an undesirable interaction with $\mathcal{K}$ and which leads to an inconsistent problem.

In case of inconsistency, a natural question for revising $\mathcal{K}$ is which of TBox or ABox axioms should be first removed. More precisely: PI assertions, NI assertions or membership assertions should be first removed. Before defining our strategy of revision, let us first point out that an inconsistency problem in *DL-Lite* is always defined with respect to some ABox, since a TBox may be incoherent but never inconsistent. We remind the Calvanese *et al.* result (Calvanese et al. 2010).

**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 $\mathcal{A}_0 \subseteq \mathcal{A}$ with at most two elements, such that $\mathcal{T} \cup \mathcal{A}_0$ is inconsistent.*

In this paper, the revision leads to ignoring some assertional facts, namely we give a priority to TBox over ABox. Furthermore we only focus on inconsistency and assume that $\mathcal{T}$ is coherent. This is not a restriction. This particular case can be handled outside the revision problem considered in this paper. Note that this choice is motivated by the fact that such situation is widely occurring in recent web applications (such as in Ontology-Based based Access applications) where the TBox is seen as a well-formed and coherent ontology whereas the ABox represent data that are not necessarily consistent with the ontology (for instance, the data that come from different sources of information by web crawling).

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

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

**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* $\mathcal{A}$=$\{B_1(a), B_3(a), B_2(b), B_3(b), B_1(c)\}$. *By Definition 3, one can easily check that* $\mathcal{C}(\mathcal{K})$=$\{\{B_1(a), B_3(a)\}, \{B_2(b), B_3(b)\}\}$.

Within the *DL-Lite* framework, in order to restore consistency while keeping new information, the Removed Sets Revision strategy removes exactly one ABox assertion in each conflict minimizing, with respect to cardinality criterion, the set of ABox assertions to be removed. Using cardinality criterion instead of set inclusion one, will reduce the set of potential conflicts. This criterion has not been considered for revising or repairing *DL-Lite* knowledge base (*e.g.* (Lembo et al. 2010; Kharlamov and Zheleznyakov 2011; Bienvenu 2012; Calvanese et al. 2010; Bienvenu and Rosati 2013)).

## Revision by a membership assertion

We now more formally present RSR according to the nature of the input information. We first consider the case where $N$ is a membership assertion that corresponds to the revision by a fact or by an observation. In the following, $\mathcal{K} \cup \{N\}$ denotes $\langle\mathcal{T}, \mathcal{A} \cup \{N\}\rangle$. The following definition introduces the concept of removed set.

**Definition 4** *Let* $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$ *be a consistent knowledge base and* $N$ *be a membership assertion. A 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* $|X| < |X'|$.

We denote by $\mathcal{R}(\mathcal{K} \cup \{N\})$ the set of the removed sets of $\mathcal{K} \cup \{N\}$. Note that if $\mathcal{K} \cup \{N\}$ is consistent then $\mathcal{R}(\mathcal{K} \cup \{N\}) = \emptyset$.

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

Indeed, assume that there are two removed sets by the Definition 4 and Definition 3, Proposition 1 leads to a contradiction.

**Definition 5** *Let* $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$ *be a consistent knowledge base and* $N$ *be a membership assertion. The revised knowledge base* $\mathcal{K} \circ_{RSR} N$ *is such that* $\mathcal{K} \circ_{RSR} N = \langle\mathcal{T}, \mathcal{A} \circ_{RSR} N\rangle$ *where* $\mathcal{A} \circ_{RSR} N = (\mathcal{A}\backslash X) \cup \{N\}$ *with* $X \in \mathcal{R}(\mathcal{K} \cup \{N\})$.

**Example 2** *Let* $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$ *be a consistent knowledge base such that* $\mathcal{T}$=$\{B_1 \sqsubseteq B_2, B_2 \sqsubseteq \neg B_3, B_3 \sqsubseteq \neg B_4\}$ *and* $\mathcal{A} = \{B_1(a), B_4(a), B_3(b)\}$. *Let* $N$=$B_3(a)$ *be a new membership assertion. One can check that* $\mathcal{K} \cup \{N\}$ *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 4* $\mathcal{R}(\mathcal{K} \cup \{N\}) = \{\{B_1(a), B_4(a)\}\}$. *Therefore* $\mathcal{A} \circ_{RSR} N = \{B_3(b), B_3(a)\}$.

As detailed in the next Section computing the set of conflicts is polynomial. Moreover when the input information is a membership assertion, as stated by Proposition 1 and illustrated in the above example, there is only one removed set.

## Revision by a positive or a negative assertion

We now consider the case where the new information $N$ is a PI assertion or a NI assertion. 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 knowledge base and* $N$ *be a PI or a NI assertion. A 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* $|X| < |X'|$.

Let us point out that Definition 6 is similar to Definition 4, except that the new information 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 should be removed. We denote again by $\mathcal{R}(\mathcal{K} \cup \{N\})$ the set of removed sets of $\mathcal{K} \cup \{N\}$.

**Example 3** *Let* $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$ *be a consistent knowledge base such that* $\mathcal{T}$=$\{B_1 \sqsubseteq B_2, B_3 \sqsubseteq \neg B_4\}$ *and* $\mathcal{A}$=$\{B_1(a), B_3(a), B_2(b), B_3(b)\}$. *Let* $N$=$B_2 \sqsubseteq \neg B_3$ *be a new assertion. One can easily check that* $\mathcal{K} \cup \{N\}$ *is inconsistent. By Definition 3,* $\mathcal{C}(\mathcal{K} \cup \{N\}) = \{\{B_1(a), B_3(a)\}, \{B_2(b), B_3(b)\}\}$ *and by Definition 6,* $\mathcal{R}(\mathcal{K} \cup \{N\}) = \{\{B_1(a), B_2(b)\}, \{B_1(a), B_3(b)\}, \{B_3(a), B_2(b)\}, \{B_3(a), B_3(b)\}\}$.

In the previous subsection, when the input is a membership assertion then there exists exactly one removed set. The situation is different when the input information is a terminological assertion, namely, a positive or a negative assertion.

Indeed, there may exist several removed sets each one leading to a possible revised knowledge base: $\mathcal{K}_i = \langle\mathcal{T} \cup \{N\}, (\mathcal{A}\backslash X_i)\rangle$ with $X_i \in \mathcal{R}(\mathcal{K} \cup \{N\})$. Within the *DL-Lite* language it is not possible to find a knowledge base that represents $\mathcal{K}_1 \vee \cdots \vee \mathcal{K}_m$ where $m = |\mathcal{R}(\mathcal{K} \cup \{N\})|$ denotes the number of removed sets.

If we want to keep the result of revision within the *DL-Lite* language several options are possible. The first one is to consider the intersection of all possible revised knowledge bases. In this case $\mathcal{K} \circ_{RSR} N$=$\langle\mathcal{T} \cup \{N\}, \mathcal{A} \circ_{RSR} N\rangle$ where $\mathcal{A} \circ_{RSR} N = \cap_{i=1}^{m}(\mathcal{A}\backslash X_i)$. Let $\phi$ be an *DL-Lite* axiom, $\phi$ is a consequence of $\mathcal{K}_1 \vee \cdots \vee \mathcal{K}_m$ if and only if $\phi$ is a consequence of each $\mathcal{K}_i$, $1 \leq i \leq m$.

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 knowledge base and* $N$ *be a PI or a NI assertion. Let* $f$ *be a selection function, the revised knowledge base* $\mathcal{K} \circ_{RSR} N$ *is such that* $\mathcal{K} \circ_{RSR} N$=$\langle\mathcal{T} \cup \{N\}, \mathcal{A} \circ_{RSR} N\rangle$ *where* $\mathcal{A} \circ_{RSR} N = (\mathcal{A}\backslash f(\mathcal{R}(\mathcal{K} \cup \{N\})))$.

Moreover, the consequences of the revised knowledge base are defined as follows.

**Definition 8** *Let* $\phi$ *be an DL-Lite axiom expressed in* $\mathcal{L}$. $\mathcal{K} \circ_{RSR} N \models \phi$ *if and only if* $\langle\mathcal{T} \cup \{N\}, \mathcal{A} \circ_{RSR} N\rangle \models \phi$.

In the next section, we provide an algorithm for computing removed sets for both revision by a membership and a PI or a NI assertion through the use of hitting sets.

## Computing the revision operation outcome

As stated before, when trying to revise a *DL-Lite* knowledge base by a membership assertion, a PI assertion or a NI assertion, we want to withdraw only ABox assertions in order to restore consistency, i.e. 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 assertions, and if the input $N$ is a PI assertion or a membership assertion, no inconsistency can occur, so the revision operation trivially becomes a simple union.

Among the remaining cases, we distinguish two different situations: (i) $N$ is a membership assertion: the computation of conflicts and the overall revision algorithm is a very simple task, thanks to Proposition 1 and will be detailed below. (ii) $N$ is a PI assertion or a NI assertions : this is the most complicated case, as several removed sets can exist. Whatever case we consider, we first need to compute the conflicts of $\mathcal{K} \cup \{N\}$.

This step follows from the algorithm given in (Calvanese et al. 2007) for checking consistency of a *DL-Lite* knowledge base. The main difference is that in (Calvanese et al. 2007) 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 pairs responsible of conflicts. Hence, we need to adapt the algorithm.

### Computing the conflicts

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 assertion 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 $\mathcal{A}'$=$\mathcal{A} \cup \{N\}$.

Computing $\mathcal{C}(\mathcal{K} \cup \{N\})$ first requires to obtain the negative closure $cln\,(\mathcal{T}')$, using the rules recalled in the refresher on *DL-lite* logic Section. 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 assertion in $cln\,(\mathcal{T}')$ in order to exhibit whether $\mathcal{A}'$ contains pairs of assertions that contradict the NI assertions. 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 $\mathcal{A}'$. Note $X(x)$ (*resp.* $Y(x)$) a basic concept, or of the form $R(x, y)$ if $X = \exists R$ (*resp.* $Y = \exists R$) or $R(y, x)$ if $X = \exists R^-$ (*resp.* $Y = \exists R^-$). The result of the evaluation of a NI assertion is a collection of sets containing two elements, or one element if $N$ is a membership assertion).

We now provide the algorithm COMPUTECONFLICTS, which computes $\mathcal{C}(\mathcal{K} \cup \{N\})$.

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

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

The set $\mathcal{C}(\mathcal{K}')$ stores the conflict sets. The first step of the algorithm consists in computing the negative closure of $\mathcal{T}'$. Then, for each NI assertion $X \sqsubseteq \neg Y$ of $cln\,(\mathcal{T}')$ 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 an empty set.

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

### Removed sets computation

When the input $N$ is a membership assertion (namely a fact), then, thanks to Proposition 1, we know that there is only one removed set.

The computation of this only removed set amounts in picking in each conflict the membership assertion which is different from the new information $N$. One can easily check that every conflict set $\{\alpha_t, \alpha_j\}$ that contradicts a NI assertion is of the form $\{x, N\}$ where $x \in \mathcal{A}$. This means that there exists exactly one removed set obtained. Hence, in this case the removed set computation can be performed in polynomial time: when returning from the call to COMPUTECONFLICTS, the only removed set is $\bigcup_{c_i \in \mathcal{C}(\mathcal{K} \cup \{N\})} (c_i \setminus \{N\})$.

Now, we detail the case where $N$ is a NI assertion. We follow the idea proposed in (Würbel, Jeansoulin, and Papini 2000), where removed sets can be computed using the hitting set notion (Reiter 1987).

A hitting set is a set which intersects each set in a collection. A minimal hitting set, with respect to set inclusion, is called a kernel. Moreover, kernels which are minimal according to cardinality correspond to the definition of a removed set. More formally, the following proposition holds:

**Proposition 2** *$R \subseteq \mathcal{A}$ is a removed set for $\mathcal{K} \cup \{N\}$ if and only if $R$ is a minimal kernel with respect to cardinality of $\mathcal{C}(\mathcal{K} \cup \{N\})$.*

The computation of hitting sets is performed using Reiter's algorithm (Reiter 1987), modified in (Wilkerson, Greiner, and Smith 1989). Moreover, as we are looking for minimal cardinality hitting sets, we can use the optimized version found in (Würbel, Jeansoulin, and Papini 2000). The computation relies on the construction of a tree in a breadth-first order. We recall this algorithm:

**Definition 9** *Let $\mathcal{O}$ be a collection of sets, a tree T is an HS-tree if and only if it is the smallest tree having the following properties:*

1. *Its root is labeled by an element from $\mathcal{O}$. If $\mathcal{O}$ is empty, its root labeled is by '$\surd$'.*
2. *If $n$ is a node from T, let $H(n)$ be the set of branch labels on the path going from the root to T to n. If $n$ is labeled by '$\surd$', it has no successor in T.*
3. *If $n$ is labeled by a set $O \in \mathcal{O}$, then, for each $o \in O$, $n$ has a successor node $n_o$ in T, joined to $n$ by a branch labeled by o. The label of $n_o$ is a set $O^{'} \in \mathcal{O}$ such that $O^{'} \cap H(n_o) \neq \emptyset$, if such a set exists. Otherwise, $n_o$ is labeled by '$\surd$'.*

*The minimal hitting sets correspond to the leaves labelled by $\surd$. For each such node $n$, $H(n)$ is a minimal hitting set of $\mathcal{O}$. We use the same pruning techniques as in (Wilkerson, Greiner, and Smith 1989), and, as in (Würbel, Jeansoulin, and Papini 2000), we stop the tree construction at the first level where we find a leaf labelled with $\surd$, since it ensures the minimal cardinality property.*

**Example 4** *Consider $\mathcal{K}$=$\langle\mathcal{T},\mathcal{A}\rangle$, with $\mathcal{T}$=$\{A \sqsubseteq \neg B, A \sqsubseteq C\}$ and $\mathcal{A}$=$\{A(a), D(a), C(a), C(b), D(b)\}$. We want to revise this base with $N$=$C \sqsubseteq \neg D$. Then, We have $cln(\mathcal{T} \cup \{C \sqsubseteq \neg D\})$=$\{A \sqsubseteq \neg B, C \sqsubseteq \neg D, A \sqsubseteq \neg D\}$. No conflicts are obtained from $A \sqsubseteq \neg B$, the conflicts obtained from $C \sqsubseteq \neg D$ are $\{D(a), C(a)\}$ and $\{D(b), C(b)\}$, and the only conflict obtained from $A \sqsubseteq \neg D$ is $\{A(a), D(a)\}$. The kernels are $\{A(a), C(a), C(b)\}$, $\{A(a), C(a), D(b)\}$, $\{D(a), C(b)\}$, $\{D(a), D(b)\}$. Among them, those which are minimal w.r.t.cardinality, and thus are the removed sets, are: $\mathcal{R}(\mathcal{K} \cup \{N\}) = \{\{D(a), C(b)\}, \{D(a), D(b)\}\}$.*

## Removed Sets Revision: logical properties

In this section we go one step further in the definition of Removed Sets Revision, by presenting logical properties of RSR operators through a set of postulates.

As mentioned in the Introduction, the AGM postulates (Alchourrón, Gärdenfors, and Makinson 1985) have been formulated to characterize belief revision. Since these postulates have been proposed for 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 (Flouris, Plexousakis, and Antoniou 2004; 2005; 2006). Indeed, the problem is that AGM postulates are defined for belief sets, i.e deductively closed sets of formulas, possibly infinite. Qi et al. (Qi, Liu, and Bell 2006) have focused on revising a finite representation of belief sets where they used a semantic reformulation of AGM postulates by Katsuno and Mendelzon (Katsuno and Mendelzon 1991) to extend it to DLs knowledge bases. However, as pointed out in (Calvanese et al. 2010) 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 (Hansson 2008) since infinite structures are cognitively inaccessible. Revision within the framework of Description logics, 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 (Fuhrmann 1997; Hansson 1998).

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

Let $\mathcal{K}$, $\mathcal{K}'$ be DL-Lite knowledge bases, $N$ and $M$ be either membership assertions or positive or negative assertions, $\circ$ be a revision operator. $\mathcal{K}$+$N$ denotes the non closing expansion, i.e. $\mathcal{K} + N = \mathcal{K} \cup \{N\}$.

| | |
|---|---|
| *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 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. |
| *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 incon--sistent. |

*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 the 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. *Core-retainment* and *Relevance* express the intuition that nothing is removed from the original knowledge bases unless its removal in some way contributes to make the result consistent.

According to these postulates, the following propositions hold.

**Proposition 3** *Let $\mathcal{K}$ be a consistent DL-Lite knowledge base and $N$ be a membership assertion. The revision operator $\circ_{RSR}$ such that $\mathcal{K} \circ_{RSR} N = \langle\mathcal{T}, \mathcal{A} \circ_{RSR} N\rangle$ satisfies* Success, Inclusion, Consistency, Vacuity, Pre-expansion, Internal exchange, Core retainment *and* Relevance.

This proposition states that RSR with a membership assertion as input satisfies all postulates. This proposition holds because in this case there exists only one removed set.

**Proposition 4** *Let $\mathcal{K}$ be a consistent DL-Lite knowledge base and $N$ be a positive or a negative assertion. The revision operator $\circ_{RSR}$ such that $\mathcal{K} \circ_{RSR} N = \langle \mathcal{T} \cup \{N\}, \mathcal{A} \circ_{RSR} N \rangle$ satisfies* Success*,* Inclusion*,* Consistency*,* Vacuity*,* Pre-expansion*,* Internal exchange *and* Core retainment*.*

## Related works

In (Calvanese et al. 2010), 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 a new knowledge (Wang, Wang, and Topor 2010). In this paper, we focus on revision.

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

The first difference concerns the form of the input. In our case the new information is a single PI assertion, NI assertion or membership assertion, that is a single formula. In (Calvanese et al. 2010), the input is a set of formulas.

The second difference is that in (Calvanese et al. 2010) they develop two operators whose strategy is to non derministically choose some maximal consistent subset. The first operator developed in (Calvanese et al. 2010), named $BoldEvol$, starts with the input, and incrementally and non-deterministically adds as many formulas from the closure of the knowledge base as possible. The algorithm for the computation of such set is polynomial. However, in case the input is a set of membership assertions, they give a result similar to our operator, namely, the result of the operation is uniquely defined, 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. Our removed set revision operator 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 relying on the principle of only working with explicitly formulated beliefs, we follow Hansson's point of view in (Hansson 2008).

Following this line, extensions of belief bases revision to Description logics have been proposed, however these approaches differ from ours in several aspects. In (Halaschek-wiener, Katz, and Parsia 2006), they focus on $SHOIN$ Description Logic, they extend kernel-based revision (Hansson 1994) and semi-revision operators (Hansson 1997) to $SHOIN$ 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 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 $SHOIN$. In order to lower this extra-complexity, they rely on an optimised version of the Pellet consistency checker which uses properties of the $SHOIN$ logic, allowing them to define an incremental version of their consistency checking tableau algorithm.

In (Ribeiro and Wassermann 2007), 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 that our, since they consider $SHOIN$ and $SHIF$ in order to capture all the OWL-DL and OWL-Lite languages. Our revision operator can be viewed as a restriction of the operator they define under the name *kernel revision without negation*. The restrictions are : (i) our belief bases are expressed in DL-Lite ; (ii) the minimality of the result of the incision function is defined in terms of cardinality in our case.

Our revision approach is also related to another important problem that often appears in the Ontology-Based Data Access domain. It is the problem of answering (complex or simple) queries addressed to an inconsistent knowledge base expressed in *DL-Lite*.

Recently several works (Giacomo et al. 2007; Lembo et al. 2010; Bienvenu 2012; Bienvenu and Rosati 2013) have been proposed to deal with such a problem. Those works are especially inspired by the approaches proposed in databases which are based on 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.

Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL knowledge base, several notions of ABox repair have been provided. For instance in (Lembo et al. 2010) the ABox Repair $(AR)$ is defined as a maximal subset of $\mathcal{A}$ w.r.t. set inclusion consistent with $\mathcal{T}$. The Intersection ABox Repair $(IAR)$ is the intersection of all the $AR$ repairs. While $AR$ and $IAR$ repairs only take into account explicit assertions included in the ABox, the Closed ABox Repair $(CAR)$, also considers assertions implied through the TBox, finally the Intersection Closed ABox Repair $(ICAR)$ is the intersection of all the $CAR$ repairs.

Since Removed Sets Revision only focuses on explicit assertions represented in the knowledge base, an $AR$-repair may correspond to a removed set however the converse does not hold since a removed set is defined in terms of minimality w.r.t cardinality and $AR$-repair is defined in terms on maximality w.r.t. set inclusion of subset consistent with $\mathcal{T}$. However, the approaches differ since DL knowledge base repair only deals with inconsistency handling while revision has also to take into account the input information. Moreover, algorithms are given only for $IAR$ and $ICAR$ repairs.

## Conclusion

In this paper, we investigated the extension of Removed Sets Revision to *DL-Lite* knowledge bases. We have considered three forms of incorporated information, namely, when the input is a TBox assertion or an ABox assertion. We first defined the Removed Sets Revision for *DL-Lite* knowledge bases according to the form of the input. We then proposed algorithms for pinpointing inconsistencies and computing the removed sets using the notion of hitting sets. We have also rephrased Hansson's postulates within DL-Lite framework and we studied logical properties of Removed Sets Revision operators.

A future work will first focus on the extension of Prioritized Removed Sets Revision for the case of prioritized *DL-Lite* knowledge bases. Another future work will deal with the extension of Removed Sets Fusion (Hué, Würbel, and Papini 2008), defined in a propositional setting, to the merging of *DL-Lite* knowledge bases.

## Acknowledgments

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

## References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.* 50(2):510–530.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The dl-lite family and relations. *J. Artif. Intell. Res. (JAIR)* 36:1–69.

Baader, F.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F. 2003. The description logic handbook: Theory, implementation, and applications. *Cambridge University Press*.

Benferhat, S.; Cayrol, C.; Dubois, D.; Lang, J.; and Prade, H. 1993. Inconsistency management and prioritized syntax-based entailment. In Bajcsy, R., ed., *IJCAI*, 640–647. Morgan Kaufmann.

Benferhat, S.; Ben-Naim, J.; Papini, O.; and Würbel, E. 2010. An answer set programming encoding of prioritized removed sets revision: application to gis. *Appl. Intell.* 32(1):60–87.

Bienvenu, M., and Rosati, R. 2013. New inconsistency-tolerant semantics for robust ontology-based data access. In Eiter, T.; Glimm, B.; Kazakov, Y.; and Krötzsch, M., eds., *Description Logics*, volume 1014 of *CEUR Workshop Proceedings*, 53–64. CEUR-WS.org.

Bienvenu, M. 2012. Inconsistency-tolerant conjunctive query answering for simple ontologies. In Kazakov, Y.; Lembo, D.; and Wolter, F., eds., *Description Logics*, volume 846 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Calvanese, D.; Giacomo, G. D.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005. Dl-lite: Tractable description logics for ontologies. In *Proc. of AAAI'2005*, AAAI '05, 602–607. AAAI Press / The MIT Press.

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

Calvanese, D.; Kharlamov, E.; Nutt, W.; and Zheleznyakov, D. 2010. Evolution of dl-lite knowledge bases. In Patel-Schneider, P. F.; Pan, Y.; Hitzler, P.; Mika, P.; Zhang, L.; Pan, J. Z.; Horrocks, I.; and Glimm, B., eds., *International Semantic Web Conference (1)*, volume 6496 of *LNCS*, 112–128. Springer.

Flouris, G.; Huang, Z.; Pan, J. Z.; Plexousakis, D.; and Wache, H. 2006. Inconsistencies, negations and changes in ontologies. In *AAAI*, 1295–1300. AAAI Press.

Flouris, G.; Plexousakis, D.; and Antoniou, G. 2004. Generalizing the agm postulates: preliminary results and applications. In Delgrande, J. P., and Schaub, T., eds., *NMR*, 171–179.

Flouris, G.; Plexousakis, D.; and Antoniou, G. 2005. On applying the agm theory to dls and owl. In Gil, Y.; Motta, E.; Benjamins, V. R.; and Musen, M. A., eds., *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, 216–231. Springer.

Flouris, G.; Plexousakis, D.; and Antoniou, G. 2006. On generalizing the agm postulates. In *STAIRS*, 132–143.

Fuhrmann, A. 1997. *An essay on contraction*. Stanford. California: CSLI Publications.

Gao, S.; Qi, G.; and Wang, H. 2012. A new operator for abox revision in dl-lite. In Hoffmann, J., and Selman, B., eds., *AAAI*. AAAI Press.

Giacomo, G. D.; Lenzerini, M.; Poggi, A.; and Rosati, R. 2007. On the approximation of instance level update and erasure in description logics. In *AAAI*, 403–408. AAAI Press.

Halaschek-wiener, C.; Katz, Y.; and Parsia, B. 2006. Belief base revision for expressive description logics. In *In Proc. of Workshop on OWL Experiences and Directions*.

Hansson, S. O. 1994. Kernel contraction. *Journal of Symbolic Logic* 59:845–859.

Hansson, S. O. 1997. Semi-revision. *Journal of Applied Non-classical Logic* 7:151–175.

Hansson, S. O. 1998. Revision of belief sets and belief bases. *Handbook of Defeasible Reasoning and Uncertainty Management systems* 3:17–75.

Hansson, S. O. 2008. Specified meet contraction. *Erkenntnis* 69(1):31–54.

Hué, J.; Würbel, E.; and Papini, O. 2008. Removed sets fusion: Performing off the shelf. In *Proc. of ECAI'08 (FIAI 178)*, 94–98.

Katsuno, H., and Mendelzon, A. 1991. Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52:263–294.

Kharlamov, E., and Zheleznyakov, D. 2011. Understanding inexpressibility of model-based abox evolution in dl-lite. In Barceló, P., and Tannen, V., eds., *AMW*, volume 749 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Kharlamov, E.; Zheleznyakov, D.; and Calvanese, D. 2013. Capturing model-based ontology evolution at the instance level: The case of dl-lite. *J. Comput. Syst. Sci.* 79(6):835–872.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2010. Inconsistency-tolerant semantics for description logics. In Hitzler, P., and Lukasiewicz, T., eds., *RR*, volume 6333 of *Lecture Notes in Computer Science*, 103–117. Springer.

Nonfjall, H., and Larsen, H. L. 1992. Detection of potential inconsistencies in knowledge bases. *International Journal of Intelligent Systems* 7(2):81–96.

Papini, O. 1992. A complete revision function in propositional calculus. In *ECAI*, 339–343.

Qi, G., and Du, J. 2009. Model-based revision operators for terminologies in description logics. In Boutilier, C., ed., *IJCAI*, 891–897.

Qi, G., and Yang, F. 2008. A survey of revision approaches in description logics. In Baader, F.; Lutz, C.; and Motik, B., eds., *Description Logics*, volume 353 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Qi, G.; Liu, W.; and Bell, D. A. 2006. Knowledge base revision in description logics. In Fisher, M.; van der Hoek, W.; Konev, B.; and Lisitsa, A., eds., *JELIA*, volume 4160 of *LNCS*, 386–398. Springer.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artif. Intell.* 32(1):57–95.

Ribeiro, M., and Wassermann, R. 2007. Base revision in description logics — preliminary results. In *Int. Workshop on Ontology Dynamics (IWOD-07)*.

Wang, Z.; Wang, K.; and Topor, R. W. 2010. A new approach to knowledge base revision in dl-lite. In Fox, M., and Poole, D., eds., *AAAI*. AAAI Press.

Wilkerson, R. W.; Greiner, R.; and Smith, B. A. 1989. A correction to the algorithm in reiter's theory of diagnosis. *Artificial Intelligence* 41:79–88.

Würbel, E.; Jeansoulin, R.; and Papini, O. 2000. Revision: an application in the framework of GIS. In Cohn, A. G.; Giunchiglia, F.; and Selman, B., eds., *KR*, 505–515. Morgan Kaufmann.