

Assertional Removed Sets Merging of DL-Lite Knowledge Bases

Salem Benferhat, Zied Bouraoui, Odile Papini, Eric Würbel

▶ **To cite this version:**

Salem Benferhat, Zied Bouraoui, Odile Papini, Eric Würbel. Assertional Removed Sets Merging of DL-Lite Knowledge Bases. The 13th international conference on Scalable Uncertainty Management, Dec 2019, Compiègne, France. hal-02392974

HAL Id: hal-02392974
https://amu.hal.science/hal-02392974

Submitted on 4 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Assertional Removed Sets Merging of DL-Lite Knowledge Bases

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

[1]CRIL-CNRS UMR 8188, Univ Artois, France; *{benferhat,bouraoui}@cril.univ-artois.fr*
[2]LIS-CNRS UMR 7020, Aix Marseille Univ, Université de Toulon, Marseille, France;
{papini,wurbel}@univ-amu.fr

Abstract. *DL-Lite* is a tractable family of Description Logics that underlies the *OWL-QL* profile of the ontology web language, which is specifically tailored for query answering. In this paper, we consider the setting where the queried data are provided by several and potentially conflicting sources. We propose a merging approach, called "Assertional Removed Sets Fusion" (ARSF) for merging $DL\text{-}Lite$ assertional bases. This approach stems from the inconsistency minimization principle and consists in determining the minimal subsets of assertions, called assertional removed sets, that need to be dropped from the original assertional bases in order to resolve conflicts between them. We give several merging strategies based on different definitions of minimality criteria, and we characterize the behaviour of these strategies with respect to rational properties. The last part of the paper shows how to use the notion of hitting sets for computing the assertional removed sets, and the merging outcome.

1 Introduction

In the last years, there has been an increasing use of ontologies in many application areas including query answering, Semantic Web and information retrieval. Description Logics (DLs) have been recognized as powerful formalisms 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 an application domain, and an assertional base (called *ABox*), containing assertional facts that instantiate terminological knowledge. Among Description Logics, a lot of attention was given to *DL-Lite* [12], a lightweight family of DLs specifically tailored for applications that use huge volumes of data for which query answering is the most important reasoning task. *DL-Lite* guarantees a low computational complexity of the reasoning process.

In many practical situations, data are provided by several and potentially conflicting sources, where getting meaningful answers to queries is challenging. While the available sources are individually consistent, gathering them together may lead to inconsistency. Dealing with inconsistency in query answering has received a lot of attention in recent years. For example, a general framework for inconsistency-tolerant semantics was proposed in [4, 5]. This framework considers two key notions: modifiers and inference strategies. Inconsistency tolerant query answering is seen as made out of a modifier, which transforms the original ABox into a set of repairs, i.e. subsets of the

original ABox which are consistent w.r.t. the TBox, and an inference strategy, which evaluates queries from these repairs. Interestingly enough, such setting covers the main existing works on inconsistency-tolerant query answering (see *e.g.* [2, 9, 22]). Pulling together the data provided by available sources and then applying inconsistency-tolerant query answering semantics provides a solution to deal with inconsistency. However, in this case valuable information about the sources will be lost. This information is indeed important when trying to find better strategies to deal with inconsistency during merging process.

This paper addresses query answering by merging data sources. Merging consists in achieving a synthesis between pieces of information provided by different sources. The aim of merging is to provide a consistent set of information, making maximum use of the information provided by the sources while not favoring any of them. Merging is an important issue in many fields of Artificial Intelligence [10]. Within the classical logic setting belief merging has been studied according different standpoints. One can distinguish model-based approaches that perform selection among the interpretations which are the closest to original belief bases. Postulates characterizing the rational behaviour of such merging operators, known as IC postulates, which have been proposed by Revesz [25] and improved by Konieczny and Pino Pérez [21] in the same spirit as the seminal AGM [1] postulates for revision. Several concrete merging operators have been proposed [26, 23, 21, 11, 20]. In contrast to model-based approaches, the formula-based approaches perform selection on the set of formulas that are explicitely encoded in the initial belief bases. Some of these approaches have been adapted in the context of DL-Lite [13]. Falappa et al. [14] proposed a set of postulates to characterize the behaviour of belief bases merging operators and concrete merging operators have been proposed [6, 8, 19, 24, 17, 14, 8]. Among these formula-based merging approaches, Removed Sets Fusion approach has been proposed in [17, 18] for merging propositional belief bases. This approach stems from removing a minimal subset of formulae, called removed set, to restore consistency. The minimality in Removed Sets Fusion stems from the operator used to perform merging, which can be the sum (Σ), the cardinality ($Card$), the maximum (Max), the lexicographic ordering ($GMax$). This approach has shown interesting properties: it is not too cautious and satisfies most rational IC postulates when extended to belief sets revision.

This paper studies *DL-Lite* Assertional Removed Sets Fusion (ARSF). The main motivation in considering ARSF is to take advantage of the tractability of *DL-Lite* for the merging process and the rational properties satisfied by ARSF operators. We consider in particular *DL-Lite*$_R$ as member of the *DL-Lite* family, which offers a good compromise between expressive power and computational complexity and underlies the *OWL2-QL* profile. We propose several merging strategies based on different definitions of minimality criterion, and we give a characterization of these merging strategies. The last section contains algorithms based on the notion hitting sets for computing the merging outcome.

2 Background

In this paper, we only consider *DL-Lite*$_R$, denoted by $\mathcal{L}$, which underlies *OWL2-QL*. However, results of this work can be easily generalized for several members of the *DL-Lite* family (see [3] 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:

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

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 $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ which consists of a nonempty domain $\Delta^{\mathcal{I}}$ and an interpretation function $.^{\mathcal{I}}$. The function $.^{\mathcal{I}}$ assigns to each individual a an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, to each concept C a subset $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to each role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$. Moreover, the interpretation function $.^{\mathcal{I}}$ is extended for all constructs of $DL\text{-}Lite_R$. For instance: $(\neg B)^{\mathcal{I}}$=$\Delta^{\mathcal{I}} \backslash B^{\mathcal{I}}$, $(\exists R)^{\mathcal{I}}$=$\{x \in \Delta^{\mathcal{I}} | \exists y \in \Delta^{\mathcal{I}} \, such\, that\, (x, y) \in R^{\mathcal{I}}\}$ and $(P^-)^{\mathcal{I}}$=$\{(y, x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} | (x, y) \in P^{\mathcal{I}}\}$. Concerning the TBox, we say that $\mathcal{I}$ satisfies a concept (*resp.* role) inclusion axiom, denoted by $\mathcal{I} \models B \sqsubseteq C$ (*resp.* $\mathcal{I} \models R \sqsubseteq E$), iff $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ (*resp.* $R^{\mathcal{I}} \subseteq E^{\mathcal{I}}$). Concerning the ABox, we say that $\mathcal{I}$ satisfies a concept (*resp.* role) membership assertion, denoted by $\mathcal{I} \models A(a_i)$ (*resp.* $\mathcal{I} \models P(a_i, a_j)$), iff $a_i^{\mathcal{I}} \in A^{\mathcal{I}}$ (*resp.* $(a_i^{\mathcal{I}}, a_j^{\mathcal{I}}) \in P^{\mathcal{I}}$). Finally, an interpretation $\mathcal{I}$ is said to satisfy $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ iff $\mathcal{I}$ satisfies every axiom in $\mathcal{T}$ and every assertion 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 setting: incoherence and inconsistency [7]. 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 $\mathcal{I}$ which is a model of $\mathcal{T}$, we have $C^{\mathcal{I}}$=$\emptyset$. In *DL-Lite* setting 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 $cln(\mathcal{T})$, represents the propagation of the NIs using both PIs and NIs in the TBox (see [12] for more details). Important properties have been established in [12] for consistency checking in *DL-Lite*: $\mathcal{K}$ is consistent if and only if $\langle cln(\mathcal{T}), \mathcal{A} \rangle$ is consistent. Moreover, every *DL-Lite* knowledge base with only PIs in its TBox is always satisfiable. However when $\mathcal{T}$ contains NI axioms then the *DL-Lite* knowledge base may be inconsistent and in an assertional-based approach only elements of ABoxes are removed to restore consistency [13].

3 Assertional Removed Sets Fusion

In this section, we study removed sets fusion to merge a set $\{\mathcal{A}_1, \cdots, \mathcal{A}_n\}$ of n assertional bases, representing different sources of information, linked to a *DL-lite* ontology $\mathcal{T}$. As representation formalism, we consider $\mathcal{M}_\mathcal{K} = \langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$, an MBox knowledge base where $\mathcal{M}_\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ is called an MBox. An MBox is simply a multi-set of membership assertions, where each $\mathcal{A}_i$ is an assertional base linked to $\mathcal{T}$. We assume that $\mathcal{M}_\mathcal{K}$ is coherent, i.e. $\mathcal{T}$ is coherent and for each $\mathcal{A}_i$, $1 \leq i \leq n$, $\langle \mathcal{T}, \mathcal{A}_i \rangle$ is consistent. However, the MBox $\mathcal{M}_\mathcal{K}$ may be inconsistent since the assertional bases $\mathcal{A}_i$ may be conflicting w.r.t. $\mathcal{T}$. We define the notion of conflict as a minimal inconsistent subset of $\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$, more formally:

Definition 1. *Let $\mathcal{M}_\mathcal{K}$=$\langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$ be an inconsistent MBox DL-Lite knowledge base. A conflict C is a set of membership assertions such that $i)$ $C \subseteq \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n$, $ii)$ $\langle \mathcal{T}, C \rangle$ is inconsistent, $iii)$ $\forall C'$, if $C' \subset C$ then $\langle \mathcal{T}, C' \rangle$ is consistent.*

We denote by $\mathcal{C}(\mathcal{M}_\mathcal{K})$ the collection of conflicts in $\mathcal{M}_\mathcal{K}$. Since $\mathcal{M}_\mathcal{K}$ is assumed to be finite, if $\mathcal{M}_\mathcal{K}$ is inconsistent then $\mathcal{C}(\mathcal{M}_\mathcal{K}) \neq \emptyset$ is also finite.

Within the *DL-Lite* framework, in order to restore consistency, the following definition introduces the notion of potential assertional removed set.

Definition 2. *Let $\mathcal{M}_\mathcal{K}$=$\langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$ be a MBox DL-Lite knowledge base. A* potential assertional removed set*, denoted by X, is a set of membership assertions such that i) $X \subseteq \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n$, ii) $\langle \mathcal{T}, (\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n) \backslash X \rangle$ is consistent,*
iii) $\forall X'$, if $X' \subset X \subseteq \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n$ then $\langle \mathcal{T}, (\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n) \backslash X' \rangle$ is inconsistent.

We denote by $\mathcal{PR}(\mathcal{M}_\mathcal{K})$ the set of potential assertional removed sets of $\mathcal{M}_\mathcal{K}$. If $\mathcal{M}_\mathcal{K}$ is consistent then $\mathcal{PR}(\mathcal{M}_\mathcal{K}) = \{\emptyset\}$. The concept of potential assertional removed sets is to some extent dual to the concept of repairs (maximally consistent subbase). Namely, if X is a potential assertional removed set then $(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_n) \backslash X$ is a repair, and conversly.

Example 1. Let $\mathcal{M}_\mathcal{K}$=$\langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$ be an inconsistent MBox *DL-Lite* knowledge base such that $\mathcal{T}$=$\{A \sqsubseteq \neg B, C \sqsubseteq \neg D\}$ and $\mathcal{M}_\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ where $\mathcal{A}_1 = \{A(a), C(a)\}$ $\mathcal{A}_2 = \{A(a), A(b)\}$ and $\mathcal{A}_3 = \{B(a), D(a), C(b)\}$. By Definition 1, $\mathcal{C}(\mathcal{M}_\mathcal{K}) = \{\{A(a), B(a)\}, \{C(a), D(a)\}\}$. Hence, by Definition 2, $\mathcal{PR}(\mathcal{M}_\mathcal{K}) = \{\{A(a), C(a)\}, \{A(a), D(a)\}, \{B(a), C(a)\}, \{B(a), D(a)\}\}$.

In order to cope with conflicting sources, merging aims at exploiting the complementarity between the sources providing the ABoxes, so merging strategies are necessary. These merging strategies are captured by total pre-orders on potential assertional removed sets. Let X and Y be two potential assertional removed sets, for each strategy P a total pre-order $\leq_P$ over the potential assertional removed sets is defined. $X \leq_P Y$ means that X is preferred to Y according to the strategy P. We define $<_P$ as the strict total pre-order associated to $\leq_P$ (i.e. $X <_P Y$ if and only if $X \leq_P Y$ and $Y \not\leq_P X$).

Definition 3. *Let $\mathcal{M}_\mathcal{K}$=$\langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$ be a MBox DL-Lite knowledge base. An* assertional removed set *according to the strategy P, denoted by X, is a set of membership assertions such that i) X is a potential assertional removed set of $\mathcal{M}_\mathcal{K}$; ii) there does not exist any Y such that Y is a potential assertional removed set of $\mathcal{M}_\mathcal{K}$ and $Y <_P X$.*

We denote by $\mathcal{R}_P(\mathcal{M}_\mathcal{K})$ the set of assertional removed sets according to the strategy P of $\mathcal{M}_\mathcal{K}$. If $\mathcal{M}_\mathcal{K}$ is consistent then $\mathcal{R}_P(\mathcal{M}_\mathcal{K}) = \{\emptyset\}$. The usual merging strategies sum-based (Σ), cardinality-based ($Card$), maximum-based (Max) and lexicographic ordering ($GMax$) are captured by the following total pre-orders. We denote by $s(\mathcal{M}_\mathcal{A})$ the ABox obtained from $\mathcal{M}_\mathcal{K}$ where every assertion expressed more than once is reduced to a singleton.

$(\Sigma):\ X \leq_\Sigma Y$ if $\sum_{1\leq i\leq n} \mid X \cap \mathcal{A}_i \mid \leq \sum_{1\leq i\leq n} \mid Y \cap \mathcal{A}_i \mid$.

$(Card):\ X \leq_{Card} Y$ if $|X \cap s(\mathcal{M}_\mathcal{A})| \leq |Y \cap s(\mathcal{M}_\mathcal{A})|$.

$(Max):\ X \leq_{Max} Y$ if $\max_{1\leq i\leq n} \mid X \cap \mathcal{A}_i \mid \leq \max_{1\leq i\leq n} \mid Y \cap \mathcal{A}_i \mid$.

$(GMax)$: For every potential assertional removed set X and every ABox $\mathcal{A}_i$, we define $p_X^{\mathcal{A}_i} = \mid X \cap \mathcal{A}_i \mid$. Let $L_X^{\mathcal{M}_\mathcal{A}}$ be the sequence $(p_X^{\mathcal{A}_1}, \ldots, p_X^{\mathcal{A}_n})$ sorted by decreasing order. Let X and Y be two potential assertional removed sets of $\mathcal{M}_\mathcal{K}$, $X \leq_{GMax} Y$ if $L_X^{\mathcal{M}_\mathcal{A}} \leq_{lex} L_Y^{\mathcal{M}_\mathcal{A}}$ [1].

The Σ strategy minimizes the number of assertions to remove from $\mathcal{M}_\mathcal{A}$. The $Card$ strategy attempts, similarly to Σ, to minimize the number of removed assertions. But it does not take into account assertions which are expressed several times. Note that the Σ and $Card$ strategies only differ if there are redundant assertions. The Max strategy tries to distribute to the best the assertions to be removed among to ABoxes. It tries to do so by removing the less possible assertions in the most hit ABox. The $GMax$ strategy is a lexicographic refinement of the Max strategy. Note that when there is only one source, all strategies become equivalent.

We now present assertional-based $DL\text{-}Lite_R$ merging operators. A merging operator is a function that maps an MBox $DL\text{-}Lite_R$ $\mathcal{M}_\mathcal{K}$=$\langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$ to a knowledge base $\Delta(\mathcal{M}_\mathcal{K})$= $\langle \mathcal{T}, \Delta(\mathcal{M}_\mathcal{A}) \rangle$, where the function Δ defined from $\mathcal{L} \times \ldots \times \mathcal{L}$ to $\mathcal{L}$, merges according to a strategy a multiset of assertions $\mathcal{M}_\mathcal{A}$ into a set of assertions denoted by $\Delta(\mathcal{M}_\mathcal{A})$. In the *DL-Lite* language, it is not possible to find a set of assertions which represents the disjunction of such possible merged sets of assertions. If we want to keep the result of merging in *DL-Lite*, several options are possible. The first one is to consider the intersection of all possible merged set of assertions 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 which allows us to define the family of ARSF operators. In this paper we consider the family of selection functions that select exactly one assertional removed set as follows.

Definition 4. *A selection function f is a mapping from $\mathcal{R}_P(\mathcal{M}_\mathcal{K})$ to $\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$ such that i) $f(\mathcal{R}_P(\mathcal{M}_\mathcal{K})) = X$ with $X \in \mathcal{R}_P(\mathcal{M}_\mathcal{K})$, ii) $f(\{\emptyset\}) = \emptyset$*

Definition 5. *Let $\mathcal{M}_\mathcal{K}$=$\langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$ be a MBox DL-Lite knowledge base, f be a selection function, and P be a strategy, the merged DL-Lite knowledge base, denoted by $\Delta_P^{arsf}(\mathcal{M}_\mathcal{K})$, is such that $\Delta_P^{arsf}(\mathcal{M}_\mathcal{K})$=$\left\langle \mathcal{T}, \Delta_P^{arsf}(\mathcal{M}_\mathcal{A}) \right\rangle$ where $\Delta_P^{arsf}(\mathcal{M}_\mathcal{A})$= $(\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n) \backslash f(\mathcal{R}_P(\mathcal{M}_\mathcal{K}))$.*

[1] $(X_1, \cdots, X_n) \leq_{lex} (Y_1, \cdots, Y_n)$ if $\exists i,\ 1 \leq i \leq n$, i)$X_i \leq Y_i$, $ii)$ $\forall j,\ 1 \leq j < i\ X_i = Y_i$.

Let $\mathcal{M}_\mathcal{K}$=$\langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$ be a MBox *DL-Lite* knowledge base,and $q(x)$ a query. Querying multiple data sources is performed by querying merged data sources and $\langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle \models q(x)$ amounts to $\left\langle \mathcal{T}, \Delta_P^{arsf}(\mathcal{M}_\mathcal{A}) \right\rangle \models q(x)$.

Example 2. Let $\mathcal{M}_\mathcal{K}$=$\langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$ be the MBox of Example 1. The potential assertional removed sets are $X_1 = \{A(a), C(a)\}$, $X_2 = \{A(a), D(a)\}$, $X_3 = \{B(a), C(a)\}$ and $X_4 = \{B(a), D(a)\}$. As illustated in the table below [2], we have $\mathcal{R}_\Sigma(\mathcal{M}_\mathcal{K}) = \{X_3, X_4\}$. Suppose the selection function f is such that $f(\mathcal{R}_\Sigma(\mathcal{M}_\mathcal{K})) = X_4$ we have $\Delta_\Sigma^{arsf}(\mathcal{M}_\mathcal{A}) = \{A(a), C(a), A(b), C(b)\}$. We have $\mathcal{R}_{Card}(\mathcal{M}_\mathcal{K}) = \{X_1, X_2, X_3, X_4\}$. Suppose the selection function f is such that $f(\mathcal{R}_{Card}(\mathcal{M}_\mathcal{K})) = X_1$ we have $\Delta_{Card}^{arsf}(\mathcal{M}_\mathcal{A}) = \{A(b), B(a), D(a), C(b)\}$. We have $\mathcal{R}_{Max}(\mathcal{M}_\mathcal{K}) = \{X_2, X_3\}$. Suppose the selection function f is such that $f(\mathcal{R}_{Card}(\mathcal{M}_\mathcal{K})) = X_2$ we have $\Delta_{Max}^{arsf}(\mathcal{M}_\mathcal{A}) = \{C(a), A(b), B(a), C(b)\}$. We have $\mathcal{R}_{GMax}(\mathcal{M}_\mathcal{K}) = \{X_3\}$ and $\Delta_{GMax}^{arsf}(\mathcal{M}_\mathcal{A}) = \{A(a), D(a), A(b), C(b)\}$.

X_i	$\lvert X_i \cap \mathcal{A}_1 \rvert$	$\lvert X_i \cap \mathcal{A}_2 \rvert$	$\lvert X_i \cap \mathcal{A}_3 \rvert$	Σ	$Card$	Max	$GMax$
X_1	2	1	0	3	**2**	2	210
X_2	1	1	1	3	**2**	**1**	111
X_3	1	0	1	**2**	**2**	**1**	**110**
X_4	0	0	2	**2**	**2**	2	200

4 Logical properties

Within the context of propositional logic, postulates have been proposed in order to classify reasonable belief bases merging operators [16, 15, 14][3]. In order to give logical properties of ARSF operators, we first rephrase these postulates within the DL-Lite framework, and then analyse to which extent the proposed operators satisfy these postulates for any selection function.

Let $\mathcal{M}_\mathcal{K} = \langle \mathcal{T}, \mathcal{M}_A \rangle$ and $\mathcal{M}_{\mathcal{K}'} = \langle \mathcal{T}, \mathcal{M}'_A \rangle$ be two MBox *DL-Lite* knowledge bases, let Δ be an assertional-based merging operator and $\langle \mathcal{T}, \Delta(\mathcal{M}_A) \rangle$ be the *DL-Lite* knowledge base resulting from merging, where $\Delta(\mathcal{M}_A)$ is a set of assertions. Let σ be a permutation over $\{1, \ldots n\}$, and $\mathcal{M}_A = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ be a multiset of assertions, $\overline{\sigma}(\mathcal{M}_A)$ denotes the set $\{\mathcal{A}_{\sigma(1)}, \ldots, \mathcal{A}_{\sigma(n)}\}$. We rephrase the postulates as follows:

[2] On each column the assertional removed sets are in bold

[3] We do not consider the IC postulates [21] since they apply to belief sets and not to belief bases.

Inclusion	$\Delta(\mathcal{M}_A) \subseteq \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$.
Symmetry	For any permutation σ over $\{1, \ldots n\}$, $\Delta(\overline{\sigma}(\mathcal{M}_A)) = \Delta(\mathcal{M}_A)$.
Consistency	$\langle \mathcal{T}, \Delta(\mathcal{M}_A) \rangle$ is consistent.
Congruence	If $\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n = \mathcal{A}'_1 \cup \ldots \cup \mathcal{A}'_n$ then $\Delta(\mathcal{M}_A) = \Delta(\mathcal{M}_{A'})$.
Vacuity	If $\langle \mathcal{T}, \mathcal{M}_A \rangle$ is consistent then $\Delta(\mathcal{M}_A) = \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$.
Reversion	If $\langle \mathcal{T}, \mathcal{M}_A \rangle$ and $\langle \mathcal{T}, \mathcal{M}_{A'} \rangle$ have the same minimal inconsistent sub--sets then $(\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n) \backslash \Delta(\mathcal{M}_A) = (\mathcal{A}'_1 \cup \ldots \cup \mathcal{A}'_n) \backslash \Delta(\mathcal{M}_{A'})$.
Core-retainment	If $\alpha \in \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$ and $\alpha \notin \Delta(\mathcal{M}_A)$ then there exists $\mathcal{A}'$ s. t. $\mathcal{A}' \subseteq \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$, $\mathcal{A}'$ is consistent but $\mathcal{A}' \cup \{\alpha\}$ is inconsistent.
Relevance	If $\alpha \in \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$ and $\alpha \notin \Delta(\mathcal{M}_A)$ then there exists $\mathcal{A}'$ s. t. $\Delta(\mathcal{M}_A) \subseteq \mathcal{A}' \subseteq \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$, $\mathcal{A}'$ is consistent but $\mathcal{A}' \cup \{\alpha\}$ is inconsistent.

Inclusion states that the union of the initial ABoxes is the upper bound of any merging operation. *Symmetry* establishes that all ABoxes are considered of equal importance. *Consistency* requires the consistency of the result of merging. *Congruence* requires that the result of merging should not depend on syntactic properties of the ABoxes. *Vacuity* says that if the union of the ABoxes is consistent w.r.t. $\mathcal{T}$ then the result of merging equals this union. *Reversion* says that if ABoxes have the same minimal inconsistent subsets w.r.t. $\mathcal{T}$ then the assertions erased in the respective ABoxes are the same. *Core-retainment* and *Relevance* express the intuition that nothing is removed from the original ABoxes unless its removal in some way contribute to make the result consistent.

Proposition 1. *Let $\mathcal{M}_\mathcal{K} = \langle \mathcal{T}, \mathcal{M}_A \rangle$ be a MBox DL-Lite knowledge base. For any selection function, $\forall P \in \{\Sigma, Card, Max, GMax\}$, Δ_P^{arsf} satisfies the* Inclusion*,* Symmetry*,* Consistency*,* Vacuity*,* Core-retainment *and* Relevance*. Δ_{Card}^{arsf} satisfies* Congruence *and* Reversion*, but $\forall P \in \{\Sigma, Max, GMax\}$, Δ_P^{arsf} does not satisfy* Congruence *nor* Reversion*.*

(sketch of the proof) For any selection function, by Definitions 4 and 5 , $\forall P \in \{\Sigma, Card, Max, GMax\}$, Δ_P^{arsf} satisfies *Inclusion*, *Symmetry*, *Consistency* ,*Vacuity* and *Core-retainment*.

Relevance: By Definition 5, for any selection function f, $\forall P \in \{\Sigma, Card, Max, GMax\}$, if $\alpha \in \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n$ and $\alpha \notin \Delta_P^{arsf}(\mathcal{M}_A)$ then $\alpha \in f(\mathcal{R}_P(\mathcal{M}_\mathcal{K}))$. Let $\mathcal{A}' = \Delta_P^{arsf}(\mathcal{M}_A)$, $\mathcal{A}'$ is consistent and $A' \cup \{\alpha\}$ is inconsistent since $\alpha \in f(\mathcal{R}_P(\mathcal{M}_\mathcal{K}))$ and $f(\mathcal{R}_P(\mathcal{M}_\mathcal{K})$ is an assertional removed set. By Definition 5, Δ_{Card}^{arsf} satisfies *Congruence* and *Reversion* since every assertion expressed more than once is reduced to a singleton.

We provide a counter-example for Δ_P^{arsf}, $\forall P \in \{\Sigma, Max, GMax\}$. Let $\mathcal{M}_\mathcal{K} = \langle \mathcal{T}, \mathcal{M}_A \rangle$ be an inconsistent MBox *DL-Lite* knowledge base such that $\mathcal{T} = \{A \sqsubseteq \neg B\}$ and $A_1 = \{A(a)\}$, $A_2 = \{A(b), B(a)\}$, $A_3 = \{B(a), A(b)\}$. The potential assertional removed sets are $\mathcal{PR}(\mathcal{M}_\mathcal{K}) = \{X_1, X_2, X_3, X_4\}$ with $X_1 = \{A(a), A(b)\}$, $X_2 = \{A(a), B(b)\}$, $X_3 = \{B(a), A(b)\}$, $X_4 = \{B(a), B(b)\}$ and the sets of assertional removed sets are $\mathcal{R}_\Sigma(\mathcal{M}_\mathcal{K}) = \{X_1, X_2\}$, $\mathcal{R}_{Max}(\mathcal{M}_\mathcal{K}) = \{X_1, X_2\}$ and $\mathcal{R}_{GMax}(\mathcal{M}_\mathcal{K}) = \{X_1, X_2\}$.

X_i	$\lvert X_i \cap \mathcal{A}_1 \rvert$	$\lvert X_i \cap \mathcal{A}_2 \rvert$	$\lvert X_i \cap \mathcal{A}_3 \rvert$	Σ	Max	$GMax$
X_1	1	1	0	**2**	**1**	**110**
X_2	1	0	1	**2**	**1**	**110**
X_3	0	2	1	3	2	210
X_4	0	1	2	3	2	210

Besides, let $\mathcal{M}_{\mathcal{K}'} = \langle \mathcal{T}, \mathcal{M}'_A \rangle$ be an inconsistent MBox *DL-Lite* knowledge base such that $\mathcal{T} = \{A \sqsubseteq \neg B\}$ and $A'_1 = \{A(a), B(b)\}$, $A'_2 = \{B(a)\}$, $A'_3 = \{A(a), A(a)\}$. We have $(\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3) = (\mathcal{A}'_1 \cup \mathcal{A}'_2 \cup \mathcal{A}'_3)$ and $\mathcal{PR}(\mathcal{M}_\mathcal{K}) = \mathcal{PR}(\mathcal{M}_{\mathcal{K}'})$, and the sets of assertional removed sets are $\mathcal{R}_\Sigma(\mathcal{M}_{\mathcal{K}'}) = \{X_3, X_4\}$, $\mathcal{R}_{Max}(\mathcal{M}_{\mathcal{K}'}) = \{X_3, X_4\}$ and $\mathcal{R}_{GMax}(\mathcal{M}_{\mathcal{K}'}) = \{X_3, X_4\}$.

X_i	$\lvert X_i \cap \mathcal{A}'_1 \rvert$	$\lvert X_i \cap \mathcal{A}'_2 \rvert$	$\lvert X_i \cap \mathcal{A}'_3 \rvert$	Σ	Max	$GMax$
X_1	1	0	2	3	2	210
X_2	2	0	1	3	2	110
X_3	0	1	1	**2**	**1**	**110**
X_4	1	1	0	**2**	**1**	**110**

$\forall P \in \{\Sigma, Max, GMax\}$ we have $\mathcal{R}_P(\mathcal{M}_\mathcal{K}) \neq \mathcal{R}_P(\mathcal{M}_{\mathcal{K}'})$, and there is no selection function such that $f(\mathcal{R}_P(\mathcal{M}_\mathcal{K})) \in \mathcal{R}_P(\mathcal{M}_{\mathcal{K}'})$ therefore $\Delta_P^{arsf}(\mathcal{M}_A) \neq \Delta_P^{arsf}(\mathcal{M}_{A'})$.

5 Computing ARSF Merging Outcome

We first show the one to one correspondence between potential assertional removed sets and minimal hitting sets w.r.t. set inclusion [28]. We recall that a set H is a *hitting set* of a collection of sets $\mathcal{C}$ iff $\forall C \in \mathcal{C}, C \cap H \neq \emptyset$.

Proposition 2. *Let X be such that $X \subseteq \cup_{1 \leq i \leq n} \mathcal{A}_i$. X is an potential assertional removed set of $\mathcal{M}_\mathcal{K}$ if and only if X is minimal hitting set w.r.t. set inclusion of $\mathcal{C}(\mathcal{M}_\mathcal{K})$.*

The proof is straightforward following Definition 2. Notice that the algorithm for the computation of the set of conflicts $\mathcal{C}(\mathcal{M}_\mathcal{K})$ is done in polynomial w.r.t. the size of $\mathcal{M}_\mathcal{K}$. This can be found *e.g.* in [7]. In the following, we provide a single algorithm to compute the potential assertional removed sets and the assertional removed sets according to the strategies $Card$, Σ, Max and $Gmax$. We give explanations on the different use cases of this algorithm hereafter. For a given assertional base $\mathcal{M}_\mathcal{K}$, the outcome of Algorithm 1 depends on the value of the parameter P: if $P \in \{Card, \Sigma, Max, Gmax\}$, then the result is $\mathcal{R}_P(\mathcal{M}_\mathcal{K})$. Otherwise the result is $\mathcal{PR}(\mathcal{M}_\mathcal{K})$.

Let us first focus on the computation of $\mathcal{PR}(\mathcal{M}_\mathcal{K})$. The algorithm is an adaptation of the algorithm for the computation of the minimal hitting sets w.r.t. set inclusion of a collection of sets described in [28]. It relies on the breadth-first construction of a directed acyclic graph called an *HS-dag*. An HS-dag T is a dag with labeled nodes and edges such that : (i) The root is labeled with $\emptyset$ if $\mathcal{C}(\mathcal{M}_\mathcal{K})$ is empty, otherwise it is labeled with an arbitrary element of $\mathcal{C}(\mathcal{M}_\mathcal{K})$; (ii) for each node n of T, we denote by $H(n)$ the set of edge labels on the path from n to the root of T ; (iii) The label of a node n is any set $C \in \mathcal{C}(\mathcal{M}_\mathcal{K})$ such that $C \cap H(n) = \emptyset$ if such a set exists. Otherwise n is labeled

Algorithm 1 Computes the elements of $\mathcal{R}_P(\mathcal{M}_\mathcal{K})$ or the elements of $\mathcal{PR}(\mathcal{M}_\mathcal{K})$ depending on the P parameter value.

function COMPUTE-ASSERTIONAL-RS($\mathcal{M}_\mathcal{K}$, P)
▷ P: strategy
  $\mathcal{M}_\mathcal{K} = \langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle, \mathcal{M}_\mathcal{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_n\}$
  $level \leftarrow 0$
  $label(root) \leftarrow$ an element $C \in \mathcal{C}(\mathcal{M}_\mathcal{K})$ ▷ $root$ is the root node
  $PrevQ \leftarrow \{root\}$ ▷ Queue of nodes in the previous level
  if $P \in \{\Sigma, Max, Gmax\}$ **then**
    $MinNodes \leftarrow \emptyset$ ▷ set of optimal nodes
    $MinCost \leftarrow \infty$ ▷ ∞ for Σ and Max, $\underbrace{(\infty, \ldots, \infty)}_{n \text{ times}}$ for $GMax$
  $mincard \leftarrow$ **false** ▷ used by $Card$ strategy
  while $PrevQ \neq \emptyset$ **and not** $mincard$ **do**
    $level \leftarrow level + 1$
    $CurQ \leftarrow \emptyset$
    for all $no \in PrevQ$ **do**
      if $label(no) \neq \emptyset$ **and** $label(no) \neq \boxtimes$ **then**
        $label(no) = \{\alpha, \beta\}$
        $label(left_branch(no)) \leftarrow \alpha$
        $label(right_branch(no)) \leftarrow \beta$
        $left_child(no) \leftarrow$PROCESSCHILD($\alpha, no, CurQ, \mathcal{M}_K, MinCost, MinNodes, P$)
        $right_child(no) \leftarrow$PROCESSCHILD($\beta, no, CurQ, \mathcal{M}_K, MinCost, MinNodes, P$)
        if $label(left_child(no)) = \emptyset$ **or** $label(right_child(no)) = \emptyset$ **and** $P = Card$ **then**
          $mincard \leftarrow \mathbf{true}$
    $PrevQ \leftarrow CurQ$
  if $P \notin \{\Sigma, Max, Gmax\}$ **then**
    $MinNodes \leftarrow$ all nodes labelled with $\emptyset$
  return $MinNodes$

Algorithm 2 Process a child branch of a node. Return a node (new or recycled).

function PROCESSCHILD($b_label, pa, CurQ, \mathcal{M}_\mathcal{K}, MinCost, MinNodes, P$)
▷ b_label: label of the branch to the new node
▷ pa: the parent node
▷ $CurQ$: queue of nodes already processed at the current level (input/output parameter)
▷ $MinCost$: current minimum cost (input/output parameter)
▷ $MinNodes$: set of current minimum cost nodes (input/output parameter)
▷ P: strategy
  $\mathcal{M}_\mathcal{K} = \langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$
  $\mathcal{M}_\mathcal{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_n\}$
  if $\exists n' \in CurQ$ such that $H(n') = H(pa) \cup \{b_label\}$ **then**
    $child_node \leftarrow n'$ ▷ no new node creation
  else if $\exists n' \in T$ such that $H(n') \subset H(pa) \cup \{b_label\}$ **and** $label(n') = \emptyset$ **then**
    $child_node \leftarrow$ a new node
    $label(child_node) \leftarrow \boxtimes$ ▷ this is a closed node
  else if $P \in \{\Sigma, Max, Gmax\}$ **and** COST($P, H(pa) \cup \{b_label\}$) $> MinCost$ **then**
    $child_node \leftarrow$ a new node
    $label(child_node) \leftarrow \boxtimes$ ▷ this is a closed node
  else
    $child_node \leftarrow$ a new node
    $label(child_node) \leftarrow$ an element $C \in \mathcal{C}(\mathcal{M}_\mathcal{K})$ such that $C \cap (H(pa) \cup \{b_label\}) = \emptyset$
    $CurQ \leftarrow CurQ \cup \{child_node\}$
    if $P \in \{\Sigma, Max, Gmax\}$ **and** $label(child_node) = \emptyset$ **then**
      if COST($P, H(pa) \cup \{b_label\}$) $< MinCost$ **then**
▷ Close current level nodes which are no more optimal
        for all $nopt \in MinNodes$ **do**
          $label(nopt) \leftarrow \boxtimes$
        $MinNodes \leftarrow \emptyset$
        $MinCost \leftarrow$COST($P, H(pa) \cup \{b_label\}$)
      $MinNodes \leftarrow MinNodes \cup \{child_node\}$
  return $child_node$

with $\emptyset$. Nodes labeled with $\emptyset$ are called *terminal nodes* ; (iv) If n is labeled by a set C, then for each $\alpha \in C$, n has a successor n_α, joined to n by an edge labeled by α.

In our case, the elements of $C \in \mathcal{C}(\mathcal{M}_\mathcal{K})$ are such that $|C| = 2$ (see [12]), so the HS-dag is binary. Algorithm 1 computes the potential assertional removed sets by computing the minimal hitting sets w.r.t. set inclusion of $\mathcal{C}(\mathcal{M}_\mathcal{K})$. It builds a *pruned HS-dag* in a breadth-first order, using some pruning rules to avoid a complete development of the branches. We move the processing of the left and right children nodes in a separate function (described in Algorithm 2), as it first permits to keep the algorithm short and simple, and second facilitates the extension of this algorithm to the computation of the assertional removed sets according to the different strategies.

$PrevQ$ and $CurQ$ are sets containing respectively the nodes of the previous and the current level. $label(n)$ denotes the label of a node n. In a similar way, if b is a branch, $label(b)$ represents the label of b. $left_branch(n)$ (*resp.* $right_branch(n)$) denotes the left (*resp.* right) branch under the node n. $left_child(n)$ (*resp.* $right_child(n)$) represent the left (*resp.* right) child node of the node n. The algorithm iterates the nodes of a level and tries to develop the branches under each of these nodes. The central property is that the conflict C labeling a node n is such that $C \cap H(n) = \emptyset$.

Pruning rules are applied when trying to develop the left and right branches of some parent node pa (lines 4–22 in function PROCESSCHILD, Algorithm 2). Let us briefly describe them: (i) if there exists a node n' on the same level as the currently developped child branch such that $H(n') = H(pa) \cup \{b_label\}$ (b_label being the label of the currently developed child branch), we connect the child branch to n', and there is no node creation (line 4); (ii) if there exists a node n' in the HS-dag such that $H(n') \subset H(pa) \cup \{b_label\}$ and n' is a terminal node, then the node connected to the child branch is a *closed node* (which is marked with $\boxtimes$) (line 6); (iii) otherwise the node connected to the child branch is labelled by a conflict C such that $H(pa) \cup \{b_label\} \cap C = \emptyset$. This new node is added to the current level queue.

Now we explain the aspects of the computation of the assertional removed sets according to each strategy P. *$Card$ strategy*. The $Card$ strategy is the simplest one to implement. First, observe that the level of a node n in the HS-dag is equal to the cardinality of $H(n)$. This means that if n is an end node (a node labeled with $\emptyset$), the cardinality of the corresponding minimal hitting set is $H(n)$. Thus, there is no need to continue the construction of the HS-dag, as we are only interested in hitting sets which are minimal w.r.t. cardinality. In the light of the preceding observation, The only modification of the algorithm is the use of a boolean flag $mincard$ which halts the computation at the end of the level where the first potential assertional removed set has been detected. *Σ, Max and $GMax$ strategies*. As regards these strategies, we have no guarantee that the assertional removed sets reside in the same level of the tree, as illustrated by the following example for the Σ strategy.

Example 3. Let $\mathcal{M}_\mathcal{K} = \langle \mathcal{T}, \mathcal{M}_\mathcal{A} \rangle$ be an inconsistent MBox $DL\text{-}Lite$ knowledge base such that $\mathcal{T} = \{A \sqsubseteq \neg B, C \sqsubseteq \neg B\}$, and $\mathcal{A}_1 = \{A(a)\}$, $\mathcal{A}_2 = \{C(a)\}$, $\mathcal{A}_3 = \{B(a)\}$, $\mathcal{A}_4 = \{B(a)\}$, $\mathcal{A}_5 = \{B(a)\}$. We have $\mathcal{PR}(\mathcal{M}_\mathcal{K}) = \{\{A(a), C(a)\}, \{B(a)\}\}$ and $\mathcal{R}_\Sigma(\mathcal{M}_\mathcal{K}) = \{\{A(a), C(a)\}\}$. Thus the only assertional removed set is found at level 2, while the first potential assertional removed set is found at level 1.

Similar examples can be exhibited for the Max and $GMax$ strategies. The search strategy and associated pruning techniques for Σ, Max and $Gmax$ are located in lines 9 and 16 of algorithm 2. They rely on a cost function which takes as parameters a strategy and a set S of ABox assertions. The different cost functions are defined according to the strategies, that is, given an MBox $\mathcal{M}_\mathcal{A} = \{\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n\}$: For the Σ strategy $\text{COST}(\Sigma, S)$ computes $|S \cap \mathcal{A}_1| + \ldots + |S \cap \mathcal{A}_n|$. For the Max strategy $\text{COST}(Max, S)$ computes $\max(|S \cap \mathcal{A}_1|, \ldots, |S \cap \mathcal{A}_n|)$, For the $GMax$ strategy, using $p_X^{\mathcal{A}_i} = |X \cap \mathcal{A}_i|$, $\text{COST}(GMax, S)$ computes $L_X^{\mathcal{M}_\mathcal{A}}$, which is the sequence $(p_X^{\mathcal{A}_1}, \ldots, p_X^{\mathcal{A}_n})$ sorted by decreasing lexicographic order.

The variable $MinCost$ maintains the current minimal cost. In line 9 of algorithm 2, if the cost of the current node is greater than $MinCost$, then the node is closed, as is cannot be optimal. Otherwise we create a new node, labelled with a conflict which does not intersect $H(pa) \cup \{b_label\}$. If such a label cannot be found (line 16), i.e. the current node is a terminal node then, at this point: (i) we are assured that $\text{COST}(P, H(pa) \cup \{b_label\}) \leq MinCost$, so we add the new node to the set of currently optimal nodes (line 22); (ii) if the cost of the current node is strictly less than $MinCost$, then we close all nodes currently believed to be optimal, empty the set containing them, and update $MinCost$ (lines 18– 21).

Example 4. We illustrate the operation of the algorithm with the computation of the assertional removed sets of example 2. Figure 1 depicts the HS-dag built by algorithm 1. Circled numbers shows the ordering of nodes (appart from root which is obviously the first node).

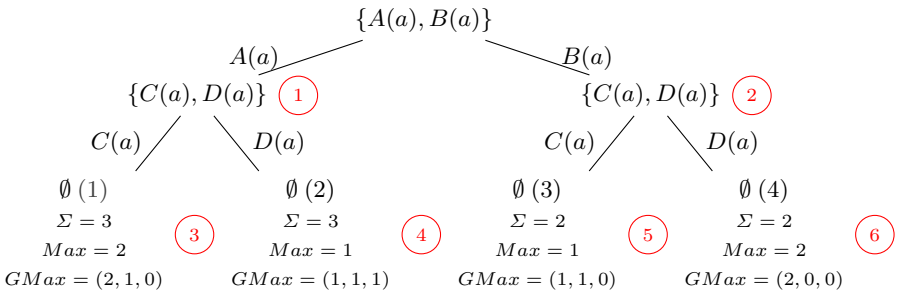


Fig. 1. Computing the removed sets of example 2.

In order to facilitate the description, we denote by $MinNodes_P$ the variable $MinNode$ when considering strategy P. The same applies for $MinCost$. At the end of the execution of the processing of a node (PROCESSCHILD function), a state of these variables is given.

root The root is labelled with a conflict.
level 1

- Left and right branches of root node are labelled respectively with $A(a)$ and $B(a)$, the members of the root label (lines 16– 17 of algorithm 1).

– PROCESSCHILD($\alpha, no, CurQ, \mathcal{M}_K, MinCost, MinNodes, P$) is called. None of the pruning conditions in lines 4, 6 and 9 apply, so node ① is created, and labelled with a conflict not intersecting $H(①) = A(a)$, namely $\{C(a), D(a)\}$. The same processing leads to the creation of node ②.
State: $MinNodes = \emptyset$, $MinCost = \infty$ for any strategy

level 2

– Left and right branches of node ① are labelled respectively with $C(a)$ and $D(a)$, the members of the label (lines 16–17 of algorithm 1).
– PROCESSCHILD($\alpha, no, CurQ, \mathcal{M}_K, MinCost, MinNodes, P$) (left branch of node ①) is called. None of the pruning conditions in lines 4, 6 and 9 apply, so node ③ is created. As there is no conflict C such that $C \cap H(③) = \emptyset$, the new node is labelled with $\emptyset$. Whatever the strategy is, its cost is necessarily less than $MinCost$ which has been initialized to ∞. Thus $MinCost$ is updated to the cost of node ③ depending on the strategy and node ③ is added to the $MinNodes$ set.
State: $MinNodes = \{③\}$, $MinCost_{\Sigma} = 3$, $MinCost_{Max} = 2$, $MinCost_{GMax} = (2, 1, 0)$.
– PROCESSCHILD($\beta, no, CurQ, \mathcal{M}_K, MinCost, MinNodes, P$) (right branch of node ①) is called. None of the pruning conditions in lines 4, 6 and 9 apply, so node ④ is created. As there is no conflict C such that $C \cap H(④) = \emptyset$, the new node is labelled with $\emptyset$. For strategy Σ, the cost of node ④ is equal to $MinCost$, thus node ④ is added to the $MinNodes$ set. For strategies Max and $GMax$, the cost of node ④ is less than $MinCost$: node ③ is closed (line 18), set $MinNodes$ is emptied, and $MinCost$ is updated.
State: $MinNodes_{\Sigma} = \{③, ④\}$, $MinNodes_{Max} = \{④\}$, $MinNodes_{GMax} = \{④\}$, $MinCost_{\Sigma} = 3$, $MinCost_{Max} = 1$, $MinCost_{GMax} = (1, 1, 1)$.
– Left and right branches of node ② are labelled respectively with $C(a)$ and $D(a)$, the members of the label (lines 16–17 of algorithm 1).
– PROCESSCHILD($\alpha, no, CurQ, \mathcal{M}_K, MinCost, MinNodes, P$) (left branch of node ②) is called. None of the pruning conditions in lines 4, 6 and 9 apply, so node ⑤ is created. As there is no conflict C such that $C \cap H(⑤) = \emptyset$, the new node is labelled with $\emptyset$. For strategy Σ, The cost of node ⑤ (2) is less than $MinCost$. The same applies for $GMax$
State: $MinNodes_{\Sigma} = \{⑤\}$, $MinNodes_{Max} = \{④, ⑤\}$, $MinNodes_{GMax} = \{⑤\}$, $MinCost_{\Sigma} = 2$, $MinCost_{Max} = 1$, $MinCost_{GMax} = (1, 1, 0)$.
– PROCESSCHILD($\beta, no, CurQ, \mathcal{M}_K, MinCost, MinNodes, P$) (right branch of node ②) is called. None of the pruning conditions apply, so node ⑥ is created. As there is no conflict C such that $C \cap H(⑥) = \emptyset$, the new node is labelled with $\emptyset$. For strategy Σ, The cost of node ⑥ (2) is equal to $MinCost$.
State: $MinNodes_{\Sigma} = \{⑤, ⑥\}$, $MinNodes_{Max} = \{④, ⑤\}$, $MinNodes_{GMax} = \{⑤\}$, $MinCost_{\Sigma} = 2$, $MinCost_{Max} = 1$, $MinCost_{GMax} = (1, 1, 0)$.

6 Conclusion

In this paper, we proposed new family of assertional-based merging operators, called Assertional Removed Sets Fusion (ARSF) operators, following several merging strate-

gies (Σ, $Card$, Max, $GMax$). We studied the behaviour of ARSF operators with respect to a set of logical postulates (initially stated for propositional formula-based merging), which we rephrased within the DL-Lite framework. From a computational point of view, we proposed algorithms, stemming from the notion of hitting set, for computing the potential assertional removed sets as well as the assertional removed sets according to the different used strategies.

Belief change has been investigated within the framework of DL-Lite. Calvanese and al. [13] adapted formula-based and model-based approaches of ABox and Tbox belief revision and update, however they did not consider belief merging. Wang and al. [27] addressed the problem of TBox DL-Lite KB merging by adapting classical model-based belief merging to DL-Lite. This approach differs from the one we propose since we extend formula-based merging to DL lite.

In a future work we plan to conduct a complexity analysis of the proposed algorithm for the different used merging strategies. Moreover, we also want to focus on the implementation of ARSF operators and on an experimental study on real world applications, in particular 3D surveys within the context of underwater archaeology and handling conflicts in dances' videos. Furthermore, the ARSF operators stem from a selection function that selects one assertional removed set, we also plan to investigate operators stemming from other selection functions as well as other strategies and other approaches than ARSF for performing assertional-based merging.

Acknowledgements: This work is partially supported by the European project H2020-MSCA-RISE: AniAge (High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage). Zied Bouraoui was supported by CNRS PEPS INS2I MODERN.

References

1. C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985.
2. M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Philadelphia, Pennsylvania, USA*, pages 68–79, 1999.
3. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The dl-lite family and relations. *J. Artif. Intell. Res. (JAIR)*, 36:1–69, 2009.
4. J. Baget, S. Benferhat, Z. Bouraoui, M. Croitoru, M. Mugnier, O. Papini, S. Rocher, and K. Tabia. A general modifier-based framework for inconsistency-tolerant query answering. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016.*, pages 513–516, 2016.
5. J. Baget, S. Benferhat, Z. Bouraoui, M. Croitoru, M. Mugnier, O. Papini, S. Rocher, and K. Tabia. Inconsistency-tolerant query answering: Rationality properties and computational complexity analysis. In *Logics in Artificial Intelligence - 15th European Conference, JELIA 2016, Larnaca, Cyprus, November 9-11, 2016, Proceedings*, pages 64–80, 2016.
6. C. Baral, S. Kraus, J. Minker, and V. S. Subrahmanian. Combining knowledge bases consisting of first order theories. *Comp. Intell.*, 8(1):45–71, 1992.
7. S. Benferhat, Z. Bouraoui, O. Papini, and E. Würbel. Assertional-based removed sets revision of DL-Lite$_R$ knowledge bases. In *ISAIM*, 2014.

8. S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Possibilistic Merging and Distance-based Fusion of Propositional Information. *Studia Logica*, 58(1):17–45, 1997.
9. M. Bienvenu. On the complexity of consistent query answering in the presence of simple ontologies. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
10. I. Bloch and A. H. (Eds). Fusion: General concepts and characteristics. *Int. J. Intell. Syst.*, 16(10):1107–1134, 2001.
11. I. Bloch and J. Lang. *Towards mathematical morpho-logics*, volume 2, pages 367–380. Springer-Verlag, 2002.
12. D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *dl-lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
13. D. Calvanese, E. Kharlamov, W. Nutt, and D. Zheleznyakov. Evolution of dl-lite knowledge bases. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm, editors, *International Semantic Web Conference (1)*, volume 6496 of *LNCS*, pages 112–128. Springer, 2010.
14. M. A. Falappa, G. Kern-Isberner, M. D. L. Reis, and G. R. Simari. Prioritized and non-prioritized multiple change on belief bases. *Journal of Philosophical Logic*, 41:77–113, 2012.
15. M. A. Falappa, G. Kern-Isberner, and G. R. Simari. Explanations, belief revision and defeasible reasoning. *Artif. Intell.*, 141(1/2):1–28, 2002.
16. A. Fuhrmann. *An essay on contraction*. CSLI Publications, Stanford. California, 1997.
17. J. Hué, O. Papini, and E. Würbel. Syntactic propositional belief bases fusion with removed sets. In *Proc. of ECSQARU'07*, pages 66–77, 2007.
18. J. Hué, E. Würbel, and O. Papini. Removed sets fusion: Performing off the shelf. In *Proc. of ECAI'08 (FIAI 178)*, pages 94–98, 2008.
19. S. Konieczny. On the difference between merging knowledge bases and combining them. In *Proc. of KR'00*, pages 135–144, 2000.
20. S. Konieczny, J. Lang, and P. Marquis. DA2 merging operators. *Artificial Intelligence*, 157:49–79, 2004.
21. S. Konieczny and R. P. Pérez. Merging information under constraints. *Journal of Logic and Computation*, 12(5):773–808, 2002.
22. D. Lembo, M. Lenzerini, R. Rosati, M. Ruzzi, and D. F. Savo. Inconsistency-tolerant query answering in ontology-based data access. *J. Web Sem.*, 33:3–29, 2015.
23. J. Lin and A. Mendelzon. Knowledge base merging by majority. In *Dynamic Worlds: From the Frame Problem to Knowledge Management*. Kluwer, 1999.
24. T. Meyer, A. Ghose, and S. Chopra. Syntactic representations of semantic merging operations. In *Proc. of PRICAI'02 (LNAI 2417)*, page 620, 2002.
25. P. Z. Revesz. On the semantics of theory change: arbitration between old and new information. *12th ACM SIGACT-SGMIT-SIGART symposium on Principes of Databases*, pages 71–92, 1993.
26. P. Z. Revesz. On the semantics of arbitration. *Journal of Algebra and Computation*, 7:133–160, 1997.
27. Z. Wang, K. Wang, Y. Jin, and G. Qi. Ontomerge a system for merging dl-lite ontologies. In *CEUR Workshop proceedings*, volume 969, pages 16–27, 2014.
28. R. W. Wilkerson, R. Greiner, and B. A. Smith. A correction to the algorithm in reiter's theory of diagnosis. *Artificial Intelligence*, 41:79–88, 1989.